

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS**

Andrei Donati

**Concepção e Desenvolvimento de um
Sistema de Recomendação para o Varejo
Físico**

Florianópolis
2018

Andrei Donati

**Concepção e Desenvolvimento de um
Sistema de Recomendação para Varejo
Físico**

Relatório submetido à Universidade Federal de Santa Catarina como requisito para a aprovação na disciplina **DAS 5511: Projeto de Fim de Curso** do curso de Graduação em Engenharia de Controle e Automação.

Orientador(a): Prof. Dr. Ricardo Rabelo

Coorientador(a): Prof. Dr. Alexandre Leopoldo Gonçalves

Florianópolis
2018

Andrei Donati

Concepção e Desenvolvimento de um Sistema de Recomendação para Varejo Físico

Esta monografia foi julgada no contexto da disciplina DAS5511: Projeto de Fim de Curso e aprovada na sua forma final pelo Curso de Engenharia de Controle e Automação.

Florianópolis, 20 de Dezembro de 2018

Banca Examinadora

Igor Marques Gois
Orientador na Empresa
Bix Tecnologia

Prof. Ricardo Rabelo
Orientador no Curso
Universidade Federal de Santa Catarina

Prof. Alexandre Gonçalves
Co-Orientador no Curso
Universidade Federal de Santa Catarina

Cleber Amaral
Avaliador
Universidade Federal de Santa Catarina

Abner do Canto Pereira
Debatedor
Universidade Federal de Santa Catarina

Angelo Baruffi Nogueira
Debatedor
Universidade Federal de Santa Catarina

AGRADECIMENTOS

Aos meus pais, que não mediram esforços para que eu chegasse até esta etapa, me proporcionando os maiores aprendizados da minha vida.

A minha namorada, pelo constante apoio e incentivo durante todo o período de faculdade, nos momentos de conquistas e tristezas.

Aos meus mentores, Igor Gois e Felipe Eberhardt que foram fundamentais para meu desenvolvimento profissional.

Aos meus amigos, pelos conselhos, estudos, conquistas e tristezas compartilhadas. Aos meus orientadores, professores Alexandre e Ricardo que me guiaram durante o todo desenvolvimento deste trabalho.

**“Sonhe grande, pois ter sonhos grandes dão o
mesmo trabalho dos sonhos pequenos.”**

Jorge Paulo Lemann

RESUMO

O setor de varejo físico é um dos mais representativos para a economia brasileira e vem enfrentando novos desafios traçados pela atual revolução digital. Neste contexto, o seguinte trabalho realiza a concepção e o desenvolvimento de um sistema de recomendação focado no setor de varejo físico. Inicialmente é levantado os requisitos deste tipo de sistema bem como as abordagens de recomendação que melhor se adequam ao cenário. Após, é desenvolvido uma proposta de arquitetura de sistema de recomendação ideal para varejo físico, baseado no método de *ensemble learning*. É realizada a implementação desta arquitetura sobre uma base de dados da empresa Grupo Soma®. Os resultados da implementação foram medidos, tendo um RMSE de 98,9% e foram considerados satisfatórios, visto que conseguiram cumprir totalmente 8 dos 11 requisitos levantados. Dessa forma, foi possível demonstrar a viabilidade da implantação de um sistema de recomendação dentro do varejo físico.

Palavras-chave: Sistema de recomendação, varejo físico, *ensemble learning*, *colaborative filtering*.

ABSTRACT

The physical retail sector is one of the most representative for the Brazilian economy and has been facing new challenges outlined by the current digital revolution. In this context, the following work conceives and develops a recommendation system focused on the physical retail sector. Initially the requirements for this type of system are raised as well as the recommendation approaches that best fit the scenario. Afterwards, a proposal of ideal recommendation system architecture for physical retail, based on the ensemble learning method, is developed. The implementation of this architecture is carried out on a database of the company Grupo Soma®. The results of the implementation were measured, with a RMSE of 98.9% and were considered satisfactory, as they were able to fully comply with 8 of the 11 requirements raised. In this way, it was possible to demonstrate the feasibility of implementing a recommendation system within the physical retail.

Key-words: Recommendation system, physical retail, *ensemble learning*, collaborative filtering

LISTA DE ILUSTRAÇÕES

Figura 1 - Charge publicada pela revista The Economist em referência a analogia de petróleo com dados	14
Figura 2 – Evolução do varejo restrito no Brasil	16
Figura 3 - Exemplos de recomendação no site Amazon.com.....	22
Figura 4 – Arquitetura genérica de um sistema de recomendação	23
Figura 5 - Recomendação de roupas no ecommerce da Hering	34
Figura 6 - Recomendações de produtos dentro do site Amazon.com	35
Figura 7 - Principais vantagens sistema de recomendação Einstein.....	36
Figura 8 - Tela criação de pipeline para entrega das recomendações	36
Figura 9 – Arquitetura do sistema proposto.....	40
Figura 10 – Arquitetura implementada sistema de recomendação.....	44
Figura 11 – Implementação da arquitetura usando PDI	44
Figura 12 – Extração dos dados com o PDI	45
Figura 13 – Parte 01 do tratamento dos dados	46
Figura 14 – Parte 02 do tratamento dos dados	47
Figura 15 – Parte 03 do tratamento dos dados	48
Figura 16 – Algoritmo de ranqueamento.....	49
Figura 17 – Algoritmo de recomendação baseada em conteúdo	50
Figura 18 – Algoritmo de filtragem colaborativa com SVD	51
Figura 19 – Exemplo de matriz Usuários x Produtos.....	52
Figura 20 – Algoritmo de escolha das recomendações	53
Figura 21 – Recomendações finais de produtos	63
Figura 22 – Recomendações finais de produtos	66

LISTA DE TABELAS

Tabela 1 – Divisão segundo o tipo de recomendação gerada	24
Tabela 2 – Divisão segundo o tipo de dado usada pelo sistema	24
Tabela 3 - Divisão segundo o tipo de abordagem utilizada.....	25
Tabela 4 - Visão geral das técnicas de filtragem colaborativa	27
Tabela 5 - Recomendações x Preferências	32
Tabela 6 – Requisitos do sistema	37
Tabela 7 – Análise das abordagens perante os requisitos.....	38
Tabela 8 – Resultado final da escolha de algoritmos.....	39
Tabela 9 - Compras realizadas pelo cliente do exemplo 1.....	61
Tabela 10 - Recomendações geradas pelo algoritmo de ranqueamento.....	61
Tabela 11 - Recomendações geradas pelo algoritmo content based	62
Tabela 12 - Recomendações geradas pelo algoritmo de filtragem colaborativa	62
Tabela 13 - Recomendações finais escolhidas	62
Tabela 14 - Compras do cliente do exemplo 01.....	63
Tabela 15 - Compras realizadas pelo cliente do exemplo 2.....	64
Tabela 16 - Recomendações geradas pelo algoritmo de ranqueamento.....	64
Tabela 17 - Recomendações geradas pelo algoritmo baseado em conteúdo	65
Tabela 18 - Recomendações geradas pelo algoritmo de filtragem colaborativa	65
Tabela 19 - Recomendações finais escolhidas	65
Tabela 20 - Compras do cliente do exemplo 02.....	66

LISTA DE ABREVIATURAS E SIGLAS

CRM – Gerenciador de Relacionamento com o Consumidor, do inglês, *Customer Relationship Management*

BI – Inteligência de Negócio, do inglês, Business Intelligence

RS – Sistema de Recomendação, do inglês, *Recommender System*

SBVC - Sociedade Brasileira de Varejo e Consumo

CB – Baseado em Conteúdo, do inglês, *Content Based*

SVM – *Support Vector Machine*

CF – Collative filtering

SAAS – *Software as a Service*

CTOR - *Click to open rate*

IA – Inteligência Artificial

PDV – Ponto de Venda

SVD – *Singular Value Decomposition*

SUMÁRIO

1	Introdução	14
1.1	Problemática.....	16
1.1.1	Pergunta de pesquisa	17
1.2	Objetivos.....	18
1.2.2	Objetivo Geral	18
1.2.3	Objetivos Específicos	18
1.3	Justificativa	18
1.4	Estrutura do documento	19
2	As empresas	20
2.1	Bix Tecnologia	20
2.2	Grupo Soma	20
3	Fundamentação Teórica	21
3.1	Sistemas de Recomendação.....	21
3.1.1	Importância dos sistemas de recomendação	21
3.1.2	Arquitetura geral dos Sistemas de Recomendação	22
3.2	Tipos de sistemas de recomendação	24
3.3.1	Ranking	26
3.3.2	Baseada em conteúdo	26
3.3.3	Baseado em demografia	27
3.3.4	Filtragem colaborativa	27
3.3.4.1	Algoritmo de similaridade	28
3.3.4.2	Uso de fatoração de matrizes.....	28
3.3.5	Baseado em utilidade.....	29
3.3.6	Baseado em conhecimento	29

3.3.7	Baseado em aprendizado de máquina	29
3.3.7.1	<i>Ensemble Learning</i>	30
3.3.8	Híbrido	30
3.4	Problemas comuns a sistema de recomendação.....	30
3.4.1	<i>Cold start</i>	31
3.4.2	Ovelha cinza	31
3.4.3	Ataques.....	31
3.5	Métricas de avaliação de sistemas de recomendação	31
3.5.1	RMSE (<i>Root Mean Squared Error</i>)	32
3.5.2	<i>Precision e Recall e False Positive Rate</i>	32
3.5.3	MAP (<i>Mean Average Precision</i>).....	33
3.6	Estudo de caso de sistemas de recomendação.....	33
3.6.1	Varejo <i>Online</i>	33
3.6.1.1	Case Chaordic	34
3.6.1.2	Amazon.....	35
3.6.2	Varejo Físico	35
4	Requisitos e Arquitetura Proposta do Sistema	37
4.1	Levantamento de requisitos	37
4.2	Análise das abordagens de recomendação	38
4.3	Escolha das abordagens a serem implementadas.....	39
4.4	Arquitetura Proposta	40
5	implementação do Sistema de recomendação	43
5.1	Base de dados utilizada	43
5.2	Ferramentas utilizadas.....	43
5.3	Implementação da arquitetura.....	44
5.3.1	Extração dos dados	45
5.3.2	Tratamento dos dados	46

5.3.3	Algoritmo de Ranqueamento	48
5.3.4	Algoritmo Baseado em Conteúdo.....	50
5.3.5	Algoritmo de Filtragem Colaborativa	51
5.3.5.1	Algoritmo de decomposição de matriz SVD	52
5.3.6	Algoritmo de escolha das recomendações.....	53
5.3.6.1	Algoritmo de <i>Machine Learning</i>	54
5.3.7	Escolha dos produtos com regras de negócio.....	54
6	Apresentação e discussão dos resultados	56
6.1	Procedimento de teste	56
6.2	Resultados Obtidos	57
6.2.1	Acuracidade	57
6.2.1.1	RMSE	57
6.2.1.2	<i>Precision</i>	58
6.2.1.3	MAP	59
6.2.2	Requisitos do sistema	59
6.3	Exemplos de recomendação obtidas.....	60
6.3.1	Exemplo 01	60
6.3.2	Exemplo 02	64
6.4	Avaliação dos Resultados	67
7	Propostas de Melhoria	68
7.1	Melhora da acurácia das recomendações	68
7.1.1	Otimização dos algoritmos de recomendação.....	68
7.1.2	Diminuição do tamanho dos grupos de produtos	68
7.1.3	Melhora do algoritmo de recomendação não personalizada	68
7.2	Outras melhorias	69
7.2.1	Implementação do treinamento <i>online</i> dos algoritmos	69

8	Considerações Finais	70
	Bibliografia	72
	APÊNDICE A – Notas dos algoritmos de recomendação.....	76
	APÊNDICE B – Algoritmo de Ranqueamento.....	78
	APÊNDICE C – Algoritmo baseado em similaridades de produto	79
	APÊNDICE D – Algoritmo de filtragem colaborativa.....	80
	APÊNDICE E – Algoritmo de Escolha final.....	81
	APÊNDICE F – Recomendação de produtos com regras de negócio.....	82

1 INTRODUÇÃO

O mundo dos negócios vem se tornando, ano a ano, mais competitivo, dinâmico e globalizado. Empresas de diferentes ramos disputam pelos diversos aspectos do consumidor, o seu dinheiro, sua atenção, suas “curtidas”, entre outros [1]. Ao mesmo tempo, dados e informações existem de maneira quase ilimitada, muitas vezes de acesso público, para quem quiser fazer o uso deles [2]. A quantidade de dados disponível vem dobrando a cada dois anos, passando 4,4 trilhões de gigabytes em 2013 para 44 trilhões de gigabytes em 2020 [3]. Portanto, o desafio é, assim como quando é o tema é petróleo, não mais em como obter a matéria prima bruta, mas sim em como transformar os dados brutos em um produto com valor comercial. A charge da figura 1 ilustra bem esta situação.

Figura 1 - Charge publicada pela revista The Economist em referência a analogia de petróleo com dados



Fonte: [4]

As maneiras que as empresas costumam usar os dados disponíveis para melhorar a sua gestão e os seus resultados são os Sistemas de Apoio a Decisão (*Business Intelligence* - BI), Gerenciador de Relacionamento com o Consumidor (*Customer Relationship Management* - CRM), Gerenciador de Relacionamento com o Fornecedor (*Supplier Relationship Management* – SRM) [5], porém estes são limitados em três aspectos:

- As análises sempre dependem de interferência humana no processo, restringindo-as a uma gama limitada de perspectivas;
- Nenhum destes sistemas consegue tomar ações palpáveis, também necessitando de uma intervenção humana e mais uma vez limitando a sua eficiência;
- Os dados analisados e tratados, na maioria esmagadora dos casos, ficam limitados a dados gerados pela empresa, deixando de lado a maior quantidade de informações;

A atual revolução digital tem trazido novas formas de aproveitar os dados disponíveis que conseguem suprir estes *gaps*, usando principalmente duas tecnologias: *Big Data* e Inteligência Artificial (IA) [6]. Técnicas de *Big Data* são usadas para que dados de várias fontes (internas e externas a empresa) consigam ser unidos, processados e utilizados. Por outro lado, a IA é usada para automatizar as análises e auxiliar nas tomadas de decisão.

Assim, uma maneira que já se provou bastante eficaz para aproveitar os dados disponíveis utilizando estas tecnologias são os Sistemas de Recomendação de Produtos que, de forma automática, conseguem oferecer sugestões de compras para os consumidores, melhorando os resultados da empresa [7].

Em contrapartida, o varejo físico é um setor que movimenta bilhões dentro do Brasil, mas que carece de soluções tecnológicas para os seus problemas (tanto os antigos problemas já bem conhecidos, como os novos que estão surgindo com a transformação digital que está em curso). Segundo [8], varejo pode ser descrito como “um conjunto de atividades de negócio que adicionam valor aos produtos e serviços vendidos aos consumidores para seu uso pessoal ou familiar”. Em termos mais práticos, varejo é uma atividade de consumo focado no consumidor final deste produto, não objetivando a revenda. Exemplos de varejistas comuns ao dia a dia de qualquer pessoa são supermercados, lojas de roupas e calçados, entre outras.

Dentro do segmento de varejo, há duas grandes divisões [9]:

- Varejo Físico: meio tradicional de distribuição dos produtos ao consumidor, usando lojas com pontos físicos, onde o consumidor consegue comprar e sair com o produto na mesma hora. Grandes exemplos deste segmento são as lojas Magazine Luiza®, Studio Z®, Walmart®, entre outros.

- Varejo Online: meio onde o consumidor visualiza e compra os produtos através de lojas virtuais (geralmente portais *web*) e recebe o produto algum tempo depois da

sua compra. Exemplos deste segmento são Netshoes®, Americanas.com®, entre outros.

O varejo representa uma grande parcela do PIB brasileiro e, com a forte concorrência industrial imposta por outros países, a vocação nacional para este setor tende a crescer. Com dados de [10], o PIB do setor de varejo restrito (varejo excluindo os setores de construção civil e setor automotivo) é de R\$1,34 trilhão, representando 20,25% do PIB nacional. Em termos de crescimento, pode-se notar altas taxas nos últimos anos (Figura 2), exceto em períodos de crise econômica.

Figura 2 – Evolução do varejo restrito no Brasil



Fonte: [10]

Desse montante, o varejo online representa modestos R\$47,7 bilhões, segundo [11]. Este valor representa aproximadamente 4% do total, ou seja, o varejo físico, apesar de estar crescendo a taxas menores que o *online*, representa a parcela principal deste setor que movimenta mais de R\$1 trilhão por ano.

1.1 Problemática

O varejo físico é um setor já bastante estudado e têm boa parte dos seus desafios já mapeados. Como exemplos destes desafios, há os custos de manutenção de pontos de vendas, administração e otimização de estoque e gerenciamento da força de trabalho. Além destes, devido a revolução digital que está ocorrendo, o varejo físico vem passando por um tempo de mudanças e novos desafios estão surgindo, como:

- A. Adoção de estratégias *omnichannel* para a integração das marcas *on* e *offline*: visto que o consumidor está cada vez mais presente no mundo digital, é uma tendência do mercado integrar marcas, fazendo com que o consumidor tenha uma experiência única, tanto no mundo *on* quanto no *offline*. Como citado por [12], o objetivo é fornecer uma experiência que possa transfigurar as possibilidades da tecnologia digital no ambiente físico. Assim, os consumidores poderiam viver a marca e sua proposta de modo mais amplo, além dos ganhos com custos operacionais que a empresa pode atingir;
- B. Concorrência com o *e-commerce*: atualmente, o varejo *online* tem tido um vertiginoso crescimento das vendas. Segundo [13], o varejo *online* cresceu 7,5% apenas nos primeiros 6 meses de 2017, com 50,3 milhões de pedidos. Vários são os motivos que explicam tal crescimento. Entre estes, [14] explica que com as ferramentas adequadas, os varejistas *online* conseguem iniciar o relacionamento com o cliente antes mesmo do fechamento da venda, estando muito mais próximo do consumidor por toda a jornada de compra por determinado produto ou serviço. Destacam-se também como motivos do crescimento do *e-commerce* a facilidade de compra e a melhoria da logística de entrega nos últimos anos;
- C. Melhorar a rentabilidade: devido aos crescentes custos necessários para construção e manutenção de um espaço físico e compra de estoque, além do aumento da concorrência com outros canais de venda, como o *e-commerce*, o varejo físico sofreu uma queda na rentabilidade [15]. Dessa forma, é necessário atuar em um dos três pilares da rentabilidade: diminuição das despesas, aumento da quantidade de clientes, aumento do valor gasto por cada cliente.

Para enfrentar estes novos desafios, novas estratégias são necessárias. Segundo [16], uma forma de enfrentar estes desafios é utilizando a coleta e análise de dados, facilitando o entendimento das necessidades, preferências e atitudes dos segmentos de consumidores. Por isso, sistemas de recomendação podem ser uma alternativa de solução aos pontos traçados.

1.1.1 Pergunta de pesquisa

Como implementar um sistema de recomendação capaz de se adaptar as regras e limitações do setor de varejo físico?

1.2 Objetivos

1.2.2 Objetivo Geral

Desenvolver um sistema de recomendação suportado por uma arquitetura baseada em *ensemble learning* voltado ao cenário do varejo físico.

1.2.3 Objetivos Específicos

- A. Levantamento dos requisitos que permitam avaliar a viabilidade de construção de um sistema de recomendação;
- B. Construção de uma arquitetura de sistema de recomendação que melhor se adeque ao cenário de varejo físico;
- C. Implementação de um sistema de recomendação e aplicação do mesmo utilizando uma base de dados real, como forma de prova de conceito;
- D. Avaliação da acuracidade do sistema de recomendação construído e aplicado sobre a base de dados real;

1.3 Justificativa

Um sistema de recomendação pode apresentar um enorme gama de benefícios, direta e indiretamente, entre os quais destacam-se [7]:

- A. Aumentar o número de vezes que um cliente visita a loja, já que produtos novos podem ser sugeridos, fomentando a curiosidade e o desejo;
- B. Tornar a experiência do cliente mais prazerosa, já que o RS consegue pré-selecionar os produtos que ele mais vai gostar;
Além disso, sistemas de recomendação podem:
- C. Aumentar o valor dos produtos comprados, já que a recomendação pode incentivar a compra de produtos mais rentáveis para a empresa;
- D. Aumentar o *cross-selling* de produtos, já que recomendações podem ser feitas de maneira que produtos complementares sejam apresentados ao consumidor.

Estes benefícios conseguem, minimizar dois dos dois dos três novos desafios do varejo físico traçados no anteriormente. Assim, sistemas de recomendação se constituem uma boa alternativa de implementação dentro de um setor de altíssima relevância para o Brasil. Por isso, se faz necessário estudar se é possível aplicar sistemas de recomendação neste ramo, e como isto deve ser feito.

A implementação de um sistema de recomendação dentro do varejo físico se torna desafiador devido aos seguintes fatores:

- A. A obtenção dos dados é dificultada, já que não necessariamente há cadastro do cliente no estabelecimento, como ocorre no varejo *online*. Desta forma, não existem dados demográficos do consumidor. Algumas empresas implementam programas de fidelidade para transpor esta dificuldade;
- B. Não há um sistema de *feedback* formal do consumidor, apenas a realização ou não de uma compra e sua possível devolução. Isto faz com que quase não existam comentários negativos ou dados implícitos como uma “quase compra”, que constituem uma excelente fonte de informação para os sistemas de recomendação;
- C. Empresas de varejo físico, muitas vezes, apresentam uma série de regras de negócio, políticas e processos complexos que precisam ser embutidas dentro das recomendações ao consumidor;
- D. A entrega da recomendação ao consumidor (ou ao usuário chave do sistema de recomendação) no momento certo é bastante desafiador, já que seria preciso abordar o cliente na hora da visita a uma loja física, por exemplo. Há maneiras de minimizar esta questão usando entregas de recomendações por canais digitais (e-mails ou mensagens de texto, por exemplo).

Por estes motivos, os sistemas de recomendação são poucos explorados e testados dentro do cenário do varejo físico [17].

1.4 Estrutura do documento

O documento é dividido em 8 capítulos, sendo este primeiro responsável pela introdução ao tema e contextualização do problema a ser resolvido. O segundo introduz as empresas as quais foram parceiras no desenvolvimento deste trabalho. A seguir, no capítulo 3 é feita a revisão bibliográfica dos sistemas de recomendação. Nos capítulos 4 e 5 é realizado o desenvolvimento do trabalho com o levantamento dos requisitos e arquitetura do sistema e a implementação da arquitetura no caso de uso apresentado, respectivamente. No capítulo 6 é feita a mensuração dos resultados e apresentado exemplos de recomendação. No sétimo capítulo é abordado propostas de melhorias e, no último, pode ser visto a conclusão do presente trabalho.

2 AS EMPRESAS

2.1 Bix Tecnologia

A empresa BIX Tecnologia é uma consultoria de Business Intelligence focado em implantação, desenvolvimento, manutenção de sistemas Qlik e Pentaho. Além de Business Intelligence, a Bix presta consultorias na área de automação de carga de dados, desenvolvimento e organização dos dados de empresas. Com foco em grandes empresas, tem como principais clientes as indústrias Portobello S.A. e Embraco, a franqueadora de varejo Portobello Shop, as varejistas CalCenter e Grupo Soma e, no ramo financeiro, o Grupo Fontes.

Atualmente com 5 anos de existência, foi fundada por Felipe Santos Eberhardt. A sua sede é localizada em Florianópolis e hoje, ela conta com uma equipe de 14 especialistas.

O orientador deste trabalho dentro da Bix é Igor Marques de Souza Gois, sócio e consultor da BIX e formado em Engenharia de Controle e Automação pela UFSC. Igor têm experiência na área de petróleo e gás e desenvolvimento de soluções de BI para varejo.

2.2 Grupo Soma

O Grupo Soma é uma empresa carioca que desenvolve, fabrica e vende peças de moda. Atualmente, conta que com mais de 4.700 funcionários e têm 178 lojas próprias em todo o Brasil, além de lojas multimarcas e e-commerce. O grupo é detentora das marcas ANIMALE® e FARM®, além da FÁBULA®, FYI®, A.BRAND®, FOXTON®, MÁS ANIMALE® e CRIS BARROS®.

Dentro do Grupo Soma, os pontos focais deste projeto foram os integrantes do SomaLabs (laboratório de inovação do grupo) Artur Lemos e João Bruno Abou Hatem de Liz.

3 FUNDAMENTAÇÃO TEÓRICA

O seguinte capítulo abordará o funcionamento das principais abordagens de sistemas de recomendação, bem como mostrará alguns cases de empresas que os utilizam.

3.1 Sistemas de Recomendação

Segundo [18], Sistemas de Recomendação (SR) são definidos como programas que tentam recomendar os itens mais adequados (produtos ou serviços) a usuários específicos (indivíduos ou empresas). Dessa forma, qualquer sistema que ofereça uma sugestão de compra ao consumidor, personalizada ou não, é considerado um sistema de recomendação. Os primeiros artigos que tratam de filtragem colaborativa advêm da metade dos anos 1990 [19], sendo que há implementações de SR de 20 anos atrás, em sites de venda de produtos e *download* de músicas, com o objetivo de incrementar a interação dos usuários com o site.

3.1.1 Importância dos sistemas de recomendação

Sistemas de Recomendação constituem uma parte importante da história do *e-commerce*. Há vários anos, gigantes do varejo *online* já trabalham utilizando os dados dos consumidores para sugerir novas compras. Dentro da varejista Amazon®, 35% do que os consumidores compram são influenciados por recomendações, e 75% do que os assinantes assistem na Netflix® vêm de recomendações [16]. Estes percentuais significam milhares de transações diárias feitas com apoio dos RS.

Ainda sobre a varejista Amazon.com, como comentado por [20], a empresa registrou um aumento de vendas de 29% durante seu segundo trimestre fiscal, em relação ao mesmo período do ano anterior (de \$9,9 bilhões para \$12,83bilhões). Muito desse crescimento tem a ver com o modo como a Amazon® integrou recomendações em quase todas as partes do processo de compra. A figura 5 retrata exemplos de recomendação feitos pela Amazon®.

Figura 3 - Exemplos de recomendação no site Amazon.com



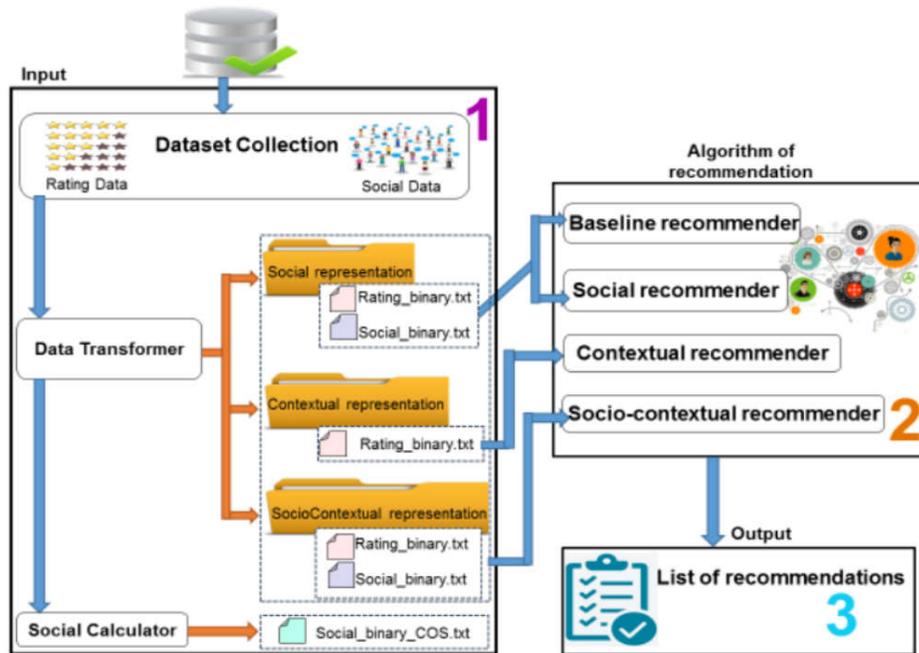
Fonte: [20]

Dessa forma, são vários os cases de sucesso de sistemas de recomendação como forma de aumentar os resultados de empresas.

3.1.2 Arquitetura geral dos Sistemas de Recomendação

Sistemas de Recomendação realizam recomendações baseadas em dados armazenados sobre os usuários, os itens a serem recomendados e as interações entre usuários e determinados itens (geralmente composta pela análise, visualização, compra e/ou comentários sobre os itens realizados pelos usuários). Com uma frequência menor os dados utilizados no processo de recomendação podem ser obtidos através de especialistas na área ou obtidos de fontes externas à empresa (dados de redes sociais, por exemplo). Além da obtenção dos dados, um sistema de recomendação apresenta mais algumas partes essenciais, como pode ser observado na Figura 6 abaixo:

Figura 4 – Arquitetura genérica de um sistema de recomendação



Fonte: [18]

As três partes referidas são:

- A. Coleta e limpeza dos dados que serão utilizados para fazer a recomendação;
- B. Implementação da abordagem de recomendação (algoritmos utilizados) usando a base de dados de “A”;
- C. Disponibilização destas recomendações conforme necessário (através de um banco de dados, de um serviço web, etc.);

O núcleo do sistema de recomendação fica na segunda parte, onde são implementados os algoritmos de recomendação. Como exemplos de algoritmos amplamente utilizados em sistemas acadêmicos e comerciais estão os pertencentes a abordagem de Filtragem Colaborativa, que se aproveita da similaridade de gostos entre diferentes pessoas em alguns assuntos, ou o algoritmo baseado em conteúdo, que busca produtos parecidos aos já consumidos para fazer recomendações. Cada um destes algoritmos apresenta pontos fortes e fracos, que serão melhor explorados a seguir.

3.2 Tipos de sistemas de recomendação

A seguir, são apresentadas 3 formas de agrupar os SR, facilitando assim o panorama geral do tema. Na tabela 1 é apresentado uma forma simples de dividir os SR [21].

Tabela 1 – Divisão segundo o tipo de recomendação gerada

Nome	Descrição	Prós e Contras
Genérica (não personalizada)	Não leva em conta qual usuário receberá a recomendação.	Prós: simples de implementar, computacionalmente fácil e não sofre do problema de <i>cold start</i> de clientes
	Leva em conta o usuário que receberá a recomendação, seja através de dados demográficos ou histórico de compras.	Contras: pouco assertivo e sofre do problema de <i>cold start</i> de produtos; Prós: muito mais assertivo e completo;
Personalizada		Contras: sofre de <i>cold start</i> sendo mais complexo de ser desenvolvido;

Fonte: Autor

Este é o modelo de divisão simples e não trata de maiores detalhes do sistema. É útil apenas como uma forma genérica de divisão dos SR. A tabela 2 agrupa dos sistemas de recomendação conforme os dados que serão utilizados como entrada para os seus algoritmos.

Tabela 2 – Divisão segundo o tipo de dado usada pelo sistema

Nome	Descrição	Exemplo
Avaliações	Feedback (explícito ou implícito) sobre o produto consumido.	Compra de um produto, avaliação após consumo, pular rapidamente uma música, etc.
Dados do produto	Características sobre o produto como gênero, cor, tipo.	Gênero da música, tamanho e cor do produto, etc.
Dados demográficos	Dados relativos ao usuário e que podem influenciar a sua decisão de consumo.	Sexo, idade, religião, preferências pessoais, restrição alimentar, etc.
Localização	Local em que o usuário se encontra ao receber a recomendação.	Em frente a uma loja de roupas, perto da seção de doces de um supermercado etc.

Fonte: Autor

Este modelo de divisão é bastante útil quando há poucos dados (ou dados muito específicos) para serem trabalhados. Dentro do varejo físico, este problema fica

evidente com a limitada quantidade e qualidade dos dados sobre os clientes e as suas avaliações. Na tabela 3 é apresentado uma divisão baseada nas abordagens de recomendação utilizadas no sistema de recomendação [22].

Tabela 3 - Divisão segundo o tipo de abordagem utilizada

Nome	Descrição	Tipo de recomendação e de dados
		Tipo de recomendação: Genérica
Ranking	Elenca os produtos seguindo uma regra (mais vendidos, maior estoque, maior rentabilidade...)	Tipo de dados: Dados do produto e avaliações
Baseado em conteúdo	É recomendado produtos com características similares aqueles que o usuário prefere ou já consumiu.	Tipo de recomendação: Personalizada Tipo de dados: Dados do produto e avaliações
Baseado em demografia	É utilizado características básicas dos usuários para fazer recomendações baseado em estereótipos e generalizações.	Tipo de recomendação: Personalizada Tipo de dados: Dados demográficos e avaliações
Filtragem Colaborativa	Busca usuários similares para extrapolar os seus gostos para o usuário que receberá a recomendação.	Tipo de recomendação: Personalizada Tipo de dados: avaliações
Baseado em utilidade	Tenta reproduzir uma função matemática de utilidade de cada recomendação	Tipo de recomendação: Personalizada Tipo de dados: Dados do produto
Baseado em conhecimento	Baseado em conhecimento explícito dos itens para fazer recomendações	Tipo de recomendação: Personalizada Tipo de dados: Dados do produto
Baseado em aprendizado de máquina	Busca reproduzir as ações de um especialista em recomendações dos produtos para os clientes.	Tipo de recomendação: Personalizada Tipo de dados: Dados do produto, demográficos e avaliações
Híbrido	Mistura 2 ou mais técnicas para formar um sistema de recomendação mais eficiente.	Tipo de recomendação: Personalizada Tipo de dados: Dados do produto, demográficos e avaliações

Fonte: Autor

A divisão por abordagem será utilizada para fazer a decisão sobre qual o tipo de sistema de recomendação pode (e deverá) ser utilizado para a construção deste

projeto, portanto será feito uma descrição mais aprofundada sobre cada abordagem a seguir.

3.3.1 Ranking

Sistemas baseados em regras são os mais simples da categoria. Produzem recomendações não personalizadas para os seus usuários baseados em regras fixas. Como explica [23] tais recomendações podem ser selecionadas manualmente, com base na popularidade dos produtos, por exemplo. Em outro exemplo, [24] cita o caso da Amazon.com®, em que caso seja visitada por um usuário anônimo apresentará os itens mais vistos por outros usuários.

Apesar de não ser muito precisa, esta abordagem pode produzir resultados aceitáveis quando há poucas informações sobre os usuários e produtos, ou quando a quantidade de produtos é baixa.

3.3.2 Baseada em conteúdo

A abordagem baseada em conteúdo parte da premissa de que há características nos itens a serem recomendados que os tornam similares entre si. Desta forma, seria viável recomendar produtos parecidos àqueles que o usuário já consumiu. As etapas básicas de um SR baseado em conteúdo são, segundo [22]:

- A. Analisar os itens preferidos pelos usuários para determinar os principais atributos comuns (preferências) que podem ser usados para distinguir esses itens;
- B. Comparar os atributos listados, de modo que apenas itens com alto grau de similaridade com o perfil do usuário sejam recomendados.

Os algoritmos utilizados por esta abordagem de recomendação são, basicamente, dois, como explicado por [22]:

- A. Usando heurística, como a medida de similaridade de cosseno;
- B. Usando aprendizado estatístico e métodos de aprendizado de máquina, construindo modelos que são capazes de aprender a similaridade entre produtos a partir de dados históricos.

Esta abordagem pode demonstrar-se bastante precisa caso os produtos não tenham características muito distintas entre si. Além disso, a escolha correta da métrica de similaridade de produtos é fundamental para a precisão e a acurácia do

sistema. Apesar disso, esta técnica é bastante impactada pelo problema de *cold start* de produtos (será tratada a frente este problema).

3.3.3 Baseado em demografia

Recomendações baseadas em demografia baseiam-se em características gerais do indivíduo e do grupo ao qual ele pertence para recomendar itens, gerando recomendações baseadas no perfil demográfico do usuário. A suposição é que diferentes recomendações devem ser geradas para diferentes nichos demográficos [25].

Um exemplo simples e comum é a recomendação de reportagens baseadas no time ao qual o usuário é torcedor. Apesar de não ser muito preciso, esta técnica garante, sem muito esforço e complexidade, que as recomendações sejam personalizadas e minimamente assertivas.

3.3.4 Filtragem colaborativa

A abordagem de filtragem colaborativa é uma das técnicas mais comuns de serem encontradas na indústria, principalmente por ser bastante precisa e não demanda uma grande quantidade de informações específicas dos produtos e/ou usuários. A Tabela 4 apresenta uma visão geral das técnicas de recomendação baseada em filtragem colaborativa.

Tabela 4 - Visão geral das técnicas de filtragem colaborativa

CF categories	Representative techniques	Main advantages	Main shortcomings
Memory-based CF	<ul style="list-style-type: none"> * Neighbor-based CF (item-based /user-based CF algorithms with Pearson/vector cosine correlation) * Item-based / user-based top-N recommendations 	<ul style="list-style-type: none"> * easy implementation * new data can be added easily and incrementally * need not consider the content of the items being recommended * scale well with co-rated items 	<ul style="list-style-type: none"> * are dependent on human ratings * performance decrease when data are sparse * can not recommend for new users and items * have limited scalability for large datasets
Model-based CF	<ul style="list-style-type: none"> * Bayesian belief nets CF * clustering CF * MDP-based CF * latent semantic CF * sparse factor analysis * CF using dimensionality reduction techniques, e.g., SVD,PCA 	<ul style="list-style-type: none"> * better address the sparsity, scalability and other problems * improve prediction performance * give an intuitive rationale for recommendations 	<ul style="list-style-type: none"> * expensive model-building * have trade-off between prediction performance and scalability * lose useful information for dimensionality reduction techniques

Hybrid recommenders	<ul style="list-style-type: none"> * content-based CF recommender, e.g., Fab * content-boosted CF * hybrid CF combining memory-based and model-based CF algorithms, e.g. Personality Diagnosis 	<ul style="list-style-type: none"> * overcome limitations of CF and content-based or other recommenders * improve prediction performance * overcome CF problems such as sparsity and gray sheep 	<ul style="list-style-type: none"> * have increased complexity and expense for implementation * need external information that usually not available
---------------------	---	--	--

Fonte: [26]

A filtragem colaborativa apresenta algumas vantagens sobre as outras abordagens, como:

- Alto nível de acurácia;
- Recomendações podem ser explicáveis para os usuários;
- Não necessita de outros dados além das avaliações dos usuários para os produtos.

3.3.4.1 Algoritmo de similaridade

A filtragem colaborativa é totalmente baseada na similaridade entre usuários e itens. Desta forma um erro em como a similaridade é mensurada invalida totalmente esta abordagem. Entre as diversas métricas de similaridade destacam-se:

- A. Cosseno: calcula o cosseno entre os vetores de avaliações dos itens pelos usuários;
- B. Cosseno ajustada: calcula o cosseno entre os vetores de avaliações dos itens pelos usuários considerando o desvio da nota atribuída ao item em relação à média das notas de determinado usuário;
- C. Jaccard: calcula a distância Jaccard entre os vetores de avaliações dos itens pelos usuários;
- D. Pearson: calcula a correlação de Pearson entre os vetores de avaliações dos itens pelos usuários;

3.3.4.2 Uso de fatoração de matrizes

O uso de fatoração de matrizes para algoritmos de recomendação ainda é incipiente devido a sua complexidade. Esta técnica funciona fazendo uma decomposição da matriz de interação entre os usuários e os produtos em matrizes menores (geralmente retangulares) [27]. Após esta decomposição, esta matriz é

reconstruída e pode ser obtido uma estimativa das avaliações dos usuários para todos os produtos (inclusive aqueles que o usuário nunca avaliou).

O principal objetivo desta técnica é diminuir o tamanho da matriz de usuário x produtos (esta matriz pode, facilmente, conter centenas de milhões de entradas) sem perder quantidade significativa de informação.

3.3.5 Baseado em utilidade

Abordagens baseadas em utilidade buscam atrelar cada recomendação a um valor de utilidade para o usuário [28]. Este tipo de abordagem não é tão utilizado devido à dificuldade de implementação em relação ao pequeno (ou nenhum) ganho de acurácia do sistema em relação a outras técnicas.

3.3.6 Baseado em conhecimento

A abordagem baseada em conhecimento busca, usando sistemas formais, representar o conhecimento dos itens e dos produtos. Com este conhecimento, é possível fazer recomendações produzindo inferências sobre as necessidades dos usuários e suas preferências [29].

Apesar de não ter uma implementação simples, já que exige a confecção de um sistema formal de representar o conhecimento, pode apresentar bons resultados principalmente com uma gama limitada de produtos.

3.3.7 Baseado em aprendizado de máquina

Esta abordagem inclui uma série de técnicas de aprendizado de máquina para prever um gosto ou preferência do usuário baseado nos dados de consumo, dos produtos ou dados demográficos dos usuários. As principais técnicas que são utilizadas para fazer a recomendação de produtos são de aprendizado bayesiano, redes neurais artificiais e Support Vector Machine (SVM). Segundo [25], essas técnicas de aprendizado de máquina são amplamente utilizadas para construir modelos de recomendação.

Esta abordagem pode apresentar grande potencial caso os dados disponíveis sejam abrangentes e precisos. Contudo, a abordagem pode facilmente se tornar complexa, dificultando ou até mesmo impossibilitando, o entendimento os motivos pela qual se atingiu determinada recomendação.

Um algoritmo geralmente utilizado na abordagem por aprendizado de máquina é o *Gradient Boosting* [30]. Ele se utiliza da técnica de *boosting* para que, com uma série de algoritmos denominados de “fracos” (*weak learners*, do inglês) conectados em série, consiga ter um bom desempenho em tarefas de classificação ou regressão. Os algoritmos utilizados, em geral, são baseados em árvores de decisão.

3.3.7.1 Ensemble Learning

A técnica de *ensemble learning* é uma técnica de aprendizado de máquina que faz o uso de vários algoritmos intermediários para obter uma performance melhorada ao final [31]. Apesar de ser uma técnica que requeira mais poder computacional, vem apresentando excelentes resultados em competições de aprendizado de máquina, principalmente quando combina algoritmos intermediários de diferentes tipos.

3.3.8 Híbrido

Abordagens híbridadas são usadas para alcançar maior desempenho e superar as desvantagens das abordagens tradicionais de recomendação ao combinar as melhores características de duas ou mais abordagens de recomendação [25]. Abordagens híbridadas, em geral, buscam combinar os benefícios dos diferentes tipos de recomendadores, principalmente no que se refere ao problema de *cold start* e escalabilidade.

Entre as abordagens híbridadas comuns destaca-se a combinação de recomendações baseadas em regras com filtragem colaborativa e aprendizado de máquina. O intuito é produzir um modelo capaz de aproveitar bem os dados disponíveis, sem sofrer tanto com problemas de cold start.

3.4 Problemas comuns a sistema de recomendação

Sistemas de recomendação, assim como qualquer outro tipo de sistema, apresentam alguns problemas que já são conhecidos e podem ser tratados. A seguir é feita uma discussão sobre os principais problemas que ocorrem em RS.

3.4.1 Cold start

O problema do *cold start* acontece quando não há dados suficientes para o algoritmo de recomendação fazer o seu trabalho de maneira adequada [32]. Este problema é visto geralmente com novos usuários (*cold start* de usuários) e novos produtos (*cold start* de produtos).

As abordagens para resolver esta questão baseiam-se em oferecer recomendações menos personalizadas e/ou trabalhar com mais de uma abordagem de recomendação, como a baseada em conteúdo e baseada em demografia. Como explicado por [33], geralmente é utilizado as abordagens de *Ranking* e Demografia para superar o problema do *cold start* de usuários. Assim, os produtos mais populares são identificados com base em tendências globais, regionais e locais ou em uma determinada hora do dia.

3.4.2 Ovelha cinza

Como explica [34], o problema identificado por “ovelha cinza” se refere aos usuários cujas opiniões não concordam consistentemente com o de outras pessoas. Deste modo, estas pessoas não se adequam aos algoritmos tradicionais de recomendação. Apesar de gerar algumas recomendações muito distante do esperado pelos usuários, em alguns casos, este tipo de erro torna-se aceitável devido à complexidade de implementar uma possível solução.

3.4.3 Ataques

Algoritmos de recomendação, assim como qualquer outro sistema, apresenta vulnerabilidades de segurança. Assim, um produto pode receber uma série de *reviews* falsos (de não compradores, do próprio vendedor do produto, entre outros) que serve unicamente para alavancar a sua recomendação. Alguns ambientes de *e-commerce* utilizam técnicas de validação do usuário para prevenir este tipo de fraude.

3.5 Métricas de avaliação de sistemas de recomendação

Tão importante quanto desenvolver um sistema de recomendação é avaliar a sua capacidade de prover sugestões adequadas considerando o contexto em que este esteja presente. Para fazer uma medição precisa de como o sistema se comporta com dados reais, é necessário ter um conjunto de dados não utilizado durante a fase de

desenvolvimento. Dessa forma, torna-se possível comparar os resultados reais com os que o RS previu. Esta técnica é chamada de *split validation*.

Além da técnica citada acima, pode-se avaliar um sistema de recomendação baseado em métricas do negócio, como o aumento das vendas, o incremento em taxas de conversão de visitantes em consumidores, a melhora do *ticket* médio da compra, entre outras métricas. O problema de mensurar um RS com base nestes fatores é que há diversas outras variáveis que influenciam nas métricas do negócio, gerando assim, muitas distorções nos resultados.

A seguir são apresentadas algumas métricas para medir a assertividade de um sistema de recomendação baseado na técnica de *split validation*.

3.5.1 RMSE (*Root Mean Squared Error*)

Esta métrica é a mais simples e geral de todas. Ela mede, de forma simplista, quanto o sistema erra em média levando-se em conta a recomendação efetuada e a avaliação realmente realizada pelo usuário.

Apesar desta métrica utilizar apenas avaliações numéricas, como notas de 1 a 5, avaliações binárias (compra ou não de um produto, “like” ou “deslike”) podem ser transformados em avaliações numéricas e utilizadas para gerar esta métrica.

3.5.2 *Precision e Recall e False Positive Rate*

As métricas de *Precision* e *Recall* buscam avaliar ocorrências de recomendações consideradas ruins, como um falso positivo ou um falso negativo [35]. Para calcular estas métricas, primeiro é estabelecida uma tabela, como a tabela 5.

Tabela 5 - Recomendações x Preferências

	Recomendado	Não recomendado
Preferido	Verdadeiro Positivo (tp)	Falso Negativo (fn)
Não preferido	Falso Positivo (fp)	Verdadeiro Negativo (tn)

Fonte: Autor

A partir da tabela as métricas são calculadas da seguinte maneira:

- *Precision*: $tp \div (tp + fp)$, mede a quantidade de falsos positivos em relação ao total de recomendações positivas apontadas pelo RS;

- *Recall* (taxa de verdadeiros positivos): $tp \div (tp + fn)$, mede qual a taxa de verdadeiros previstos em relação ao total de verdadeiros;
- *False Positive Rate* (taxa de falso positivo): $fp \div (fp + tn)$, mede a quantidade de falsos positivos em relação a quantidade total de itens não preferidos pelo usuário;

3.5.3 MAP (*Mean Average Precision*)

A métrica de MAP diferencia-se das demais por se calcular o erro diante de uma lista de recomendações [36]. É calculado segundo a fórmula:

$$MAP = \frac{1}{Q_r} \sum_{q \in Q_r} AP(q)$$

onde Q_r é o conjunto total de listas de produtos recomendados e AP é:

$$AP = \frac{1}{q_a} \sum_{i=1}^n P(i, q)$$

onde n número de recomendações da lista e q_a é o total de recomendações corretas na lista. A definição de P é:

$$P = \text{se } q = 1, \sum_{j=1}^i q_j; \text{ senão, } 0$$

onde $q = 1$ significa uma recomendação correta e $q = 0$ uma recomendação errada. Esta é uma métrica também utilizada para medir a eficiência de algoritmos de *ranking*.

3.6 Estudo de caso de sistemas de recomendação

A seguir será apresentado algumas empresas que já utilizam sistemas de recomendação.

3.6.1 Varejo Online

A seguir será apresentado caso de empresas dentro do varejo *online*.

3.6.1.1 Case Chaordic

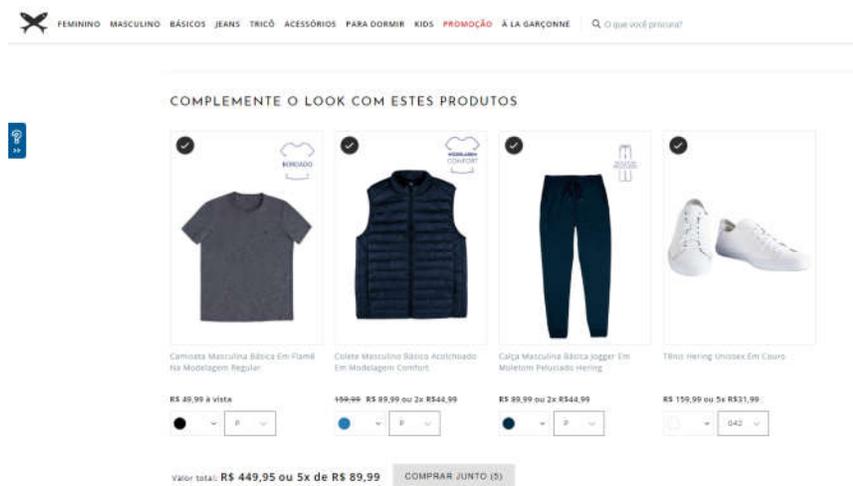
A empresa Chaordic de Florianópolis é referência em toda a América Latina no tema recomendação de produtos para e-commerce. Seu principal produto é um SAAS onde grandes varejistas online podem consumir (via uma interface de API) recomendações personalizadas para os clientes que estão navegando no seu e-commerce..

A Chaordic possui 4 grandes produtos:

- A. Sistema de busca personalizada conforme o histórico de compra do cliente;
- B. Personalização das “vitrines” do e-commerce;
- C. E-mail automatizado conforme ações do usuário e possíveis oportunidades;
- D. Recomendação de produtos por e-mail;

Um grande case de sucesso da Chaordic é no e-commerce na fabricante de roupas Hering.

Figura 5 - Recomendação de roupas no e-commerce da Hering



Fonte: [37]

Segundo dados retirados de [37], o sistema da recomendação da Chaordic conseguiu atuar nas três principais alavancas de aumento de receita do e-commerce:

- 1- Fluxo de pessoas, aumentando em 26% o CTOR em e-mails personalizados;
- 2- Conversão de visitantes em clientes, aumentando o engajamento em 3,4x nas vitrines personalizadas do e-commerce;
- 3- Ticket médio, aumentando os pedidos de 28% dos usuários comprantes.

3.6.1.2 Amazon

O sistema de recomendação da Amazon é tido como um dos mais aprimorados de todo o e-commerce, muito devido a expertise construídos em quase 20 anos de varejo online utilizando técnicas de recomendação. A Amazon chama o seu algoritmo de "matemática caseira de filtragem colaborativa item a item" e usa esse algoritmo para personalizar bastante a experiência de navegação dos clientes recorrentes [20].

O principal ponto forte deste sistema está na integração com o sistema de envio de e-mails, como explicitado em [20], que afirma que a taxa de conversão e a eficiência desses e-mails são "muito altas", significativamente mais eficazes do que as recomendações na página de compra do produto.

Além deste, vale destacar as recomendações de produtos possíveis de comprar juntos, o qual incentiva o consumidor a aumentar o seu ticket médio; e o produtos similares ao item que está sendo visitado, otimizando a descoberta de produtos.

Figura 6 - Recomendações de produtos dentro do site Amazon.com

Frequently bought together

Total price: **\$883.39**

! These items are shipped from and sold by different sellers. Show details

- This item:** Acer NH.Q2QAA.012 Nitro 5, 7th Gen Intel Core i5-7300HQ, GeForce GTX 1050 Ti, 16GB DDR4, 256GB SSD... **\$869.00**
- AmazonBasics 15.6-Inch Laptop and Tablet Bag **\$14.39**

Customers who viewed this item also viewed Page 1 of 10

Product	Price	Rating
Acer Nitro High Performance 15.6 inch Full HD Gaming Laptop Intel Core i5-7300HQ Quad-Core NVIDIA GeForce...	\$899.00 ✓ prime	2
Acer Nitro 5 Flagship Gaming VR Ready Laptop, 15.6 Inch FHD Display, Intel Quad Core i5...	\$722.90 ✓ prime	2
Acer 15.6" Nitro 5 AN515-51-72HL IPS Intel Core i7 7th Gen 7700HQ 2.8GHz NVIDIA GeForce GTX...	\$939.00 ✓ prime	4
Acer Nitro 5, 7th Gen Intel Core i5-7300HQ, GeForce GTX 1050, 8GB DDR4, 256GB SSD, Windows 10...	\$716.00	6
2018 Flagship Acer Nitro 5 15.6 Inch Full HD IPS VR Ready Gaming Laptop Intel Quad-Core i5...	\$679.00	2
2018 Flagship Premium Newest Acer Nitro 5 15.6 Inch FHD IPS Gaming Laptop (Intel Core i5...	\$669.00	5

Fonte: [39]

3.6.2 Varejo Físico

Dentro do varejo físico, sistemas de recomendação ainda não é um tema amplamente discutido, devido as várias dificuldades já citadas neste documento. A

desenvolvedora de CRM Salesforce adicionou recentemente a sua plataforma um sistema de recomendação denominado Einstein. As suas principais vantagens pode ser vista na Figura 7.

Figura 7 - Principais vantagens sistema de recomendação Einstein

AI-powered commerce is easier than you think with Einstein.

<p>No data scientist required.</p> <p>Many brands still see AI as an expensive, complex undertaking that requires a fleet of experts to implement. Salesforce has the only ecommerce solution with AI infused into the very fabric of the platform. Use Commerce Cloud's Einstein-powered capabilities to transform customer data into intelligent action in real time – no specialist required.</p>	<p>Personalization made easy.</p> <p>Today's shoppers demand meaningful digital experiences that speak directly to their individual needs. Commerce Cloud Einstein eliminates the guesswork and tailors each shopping experience as it unfolds. Deliver spot-on product recommendations, tailored product sorting, and meaningful search results at every touchpoint.</p>
<p>Always learning, always getting smarter.</p> <p>Commerce Cloud Einstein consumes and analyzes customer behavior, product data, and order history as it happens, in real time. The result? Highly relevant experiences that automatically adjust and evolve with every choice a shopper makes.</p>	<p>Time-saving capabilities.</p> <p>Commerce Cloud Einstein eliminates the need for labor-intensive tasks like manual merchandising, creating new product groupings, updating customer segments, and optimizing sorting rules. This frees your team to focus on what really matters: strategy, development, and growth.</p>

Fonte: [38]

As principais funcionalidades deste sistema é a própria integração com o CRM da empresa Salesforce e a possibilidade de criação de pipelines de abordagem do cliente com as recomendações, conforme Figura 8.

Figura 8 - Tela criação de pipeline para entrega das recomendações



Fonte: [39]

4 REQUISITOS E ARQUITETURA PROPOSTA DO SISTEMA

Como pode ser visto no capítulo anterior, há diversas abordagens e algoritmos de recomendação que podem ser utilizados para se atingir o objetivo proposto neste trabalho. Dessa forma, inicialmente será realizado um levantamento de requisitos para que seja possível definir as abordagens que melhor se adequem ao contexto do trabalho.

4.1 Levantamento de requisitos

Os requisitos a seguir foram obtidos com base em entrevistas com pessoas que trabalham na área e experiências do autor e estão ordenados conforme a sua importância na Tabela 6.

Tabela 6 – Requisitos do sistema

Ordem	Requisito	Descrição
1	Assertividade	Maior %acerto das recomendações com as compras feitas
2	Regras de negócio	Possibilidade de inserir regras específicas do negócio
3	Dados gerais de produtos	Não deve utilizar dados muito específicos dos produtos
4	Dados demográficos	Não deve utilizar dados demográficos dos clientes
5	Escalabilidade	É possível que seja escalado para milhares de clientes
6	Implementação	Menor custo e tempo de implementação
7	Estabilidade	Não deve apresentar recomendações absurdas ou que violem regras de negócio
8	<i>Feedbacks</i> negativos	Trabalhar com poucos <i>feedbacks</i> negativos
9	Recomendações explicáveis	As recomendações devem ser "explicáveis"
10	Integração	Integração com outros sistemas (PDV, CRM, entre outros)
11	<i>Cold start</i>	Lida bem com <i>cold start</i> de produtos e clientes

Fonte: Autor

Vale destacar dois requisitos:

Requisito 02 - Adaptação a regras de negócio: Este requisito explicita a necessidade do sistema respeitar as regras da empresa, como por exemplo não recomendar produtos sem estoque ou com alto *lead time*, não recomendar promoções ou produtos de coleções passadas, entre outros.

Requisito 04 – Dados demográficos: o sistema não deve se basear em dados demográficos dos clientes, visto que esses dados são de difícil obtenção dentro do varejo físico.

4.2 Análise das abordagens de recomendação

Para chegar aos algoritmos que mais se adequam ao cenário de varejo físico torna-se necessário entender quais abordagens conseguem atender cada requisito, assim como, aqueles algoritmos que não conseguem atender determinado requisito. A Tabela 7 apresenta a lista dos requisitos com as abordagens os atendem não não atendem.

Tabela 7 – Análise das abordagens perante os requisitos

Requisito	Abordagens que atendem	Abordagens que não atendem
Assertividade	Filtragem Colaborativa, Baseado em aprendizado de máquina, Baseado em conteúdo	Rankeamento
Regras de negócio	Rankeamento, Baseado em aprendizado de máquina, Baseado em conteúdo, Baseado em utilidade	
Dados gerais de produtos	Rankeamento, Baseado em conteúdo, Baseado em demografia, Baseado em aprendizado de máquina	Baseado em conteúdo
Dados demográficos	Filtragem Colaborativa, Rankeamento, Baseado em conteúdo, Baseado em aprendizado de máquina	Baseado em demografia
Escalabilidade	Rankeamento, Baseado em conteúdo, Baseado em demografia, Baseado em conhecimento	
Implementação	Filtragem Colaborativa, Rankeamento	Baseado em conhecimento
Estabilidade	Baseado em conteúdo, Rankeamento, Baseado em demografia	Baseado em aprendizado de máquina

Trabalhar com poucos feedbacks negativos	Baseado em conteúdo, Ranqueamento, Baseado em demografia	
Recomendações explicáveis	Rankeamento, Baseado em conteúdo, Baseado em demografia	Baseado em conhecimento, Baseado em utilidade, Baseado em aprendizado de máquina
Integração	Todos	
<i>Cold start</i>	Rankeamento, Baseado em conhecimento	Filtragem Colaborativa, Baseado em demografia, Baseado em conteúdo

Fonte: Autor

4.3 Escolha das abordagens a serem implementadas

A escolha das abordagens foi realizada considerando a Tabela 7, fornecendo uma “nota” inversamente proporcional à ordem do requisito, sendo positiva caso a abordagem atenda determinado requisito e negativa caso ela não atenda. Mais precisamente, a seguinte fórmula foi usada:

$$N = \sum_{r=1}^{11} \left(\frac{1}{r} \times (p) \right)$$

Onde r representa a ordem do requisito, p é 1 caso a abordagem atenda o requisito, -1 caso não atenda e 0 caso seja neutra. Por exemplo, para a abordagem de ranking, os requisitos 2 a 11 são atendidos, gerando uma nota positiva de 2,02, e o requisito 1 não é cumprido, gerando uma penalização de -1. Assim, a nota final resultante é 1,02. A nota de cada abordagem pode ser visto no APÊNDICE A – Notas dos algoritmos de recomendação.

Os resultados finais obtidos podem ser vistos na Tabela 8 e indicam as abordagens que melhor atendem aos requisitos definidos na Tabela 7.

Tabela 8 – Resultado final da escolha de algoritmos

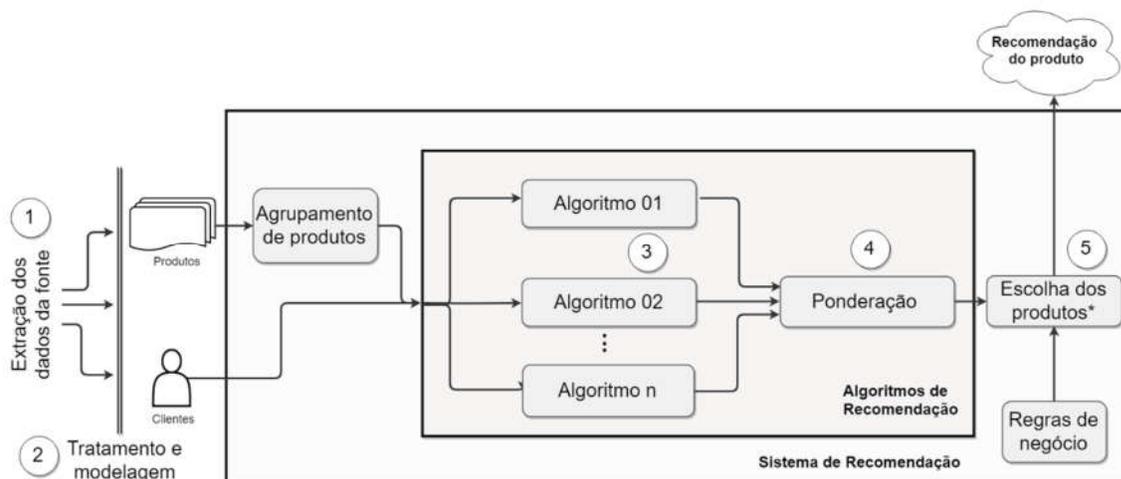
Ordem	Abordagem	Nota Obtida
1	Baseado em conteúdo	2,34
2	Baseado em aprendizado de máquina	1,93
3	Filtragem Colaborativa	1,43
4	Rankeamento	1,02
5	Baseado em demografia	0,67
6	Baseado em utilidade	0,49

Dessa forma, serão implementadas as 4 primeiras abordagens de recomendação dado que estas são aquelas que obtiveram nota superior a 1.

4.4 Arquitetura Proposta

Para melhor seguir os requisitos levantados, é proposto uma arquitetura de sistema de recomendação, conforme a Figura 9 a seguir (detalhes sobre a implementação da arquitetura são discutidos no Capítulo 5).

Figura 9 – Arquitetura do sistema proposto



Fonte: Autor

A arquitetura proposta faz o uso da técnica de *ensemble learning*, a qual utiliza dois ou mais algoritmos empilhados para gerar uma recomendação final. Dessa forma é possível transpor alguns problemas das abordagens de RS quando utilizadas isoladamente. A seguir, realiza-se uma descrição sobre cada componente da arquitetura do sistema proposto.

A. Extração dos dados

Os dados utilizados para fazer recomendações estão disponíveis, inicialmente, em bancos de dados transacionais ou analíticos. Por isso, é necessário coletar os dados destes locais e torna-los disponíveis localmente para que seja efetuado o tratamento e uma modelagem mais adequada.

Geralmente a etapa de extração é executada por um *software* específico que se conecta ao banco de dados e transporta os dados para outro banco de dados ou para arquivos locais.

B. Tratamento dos dados

Os dados extraídos estão em estado bruto, ou seja, não há regras de negócios aplicadas e a modelagem não está otimizada para servir de entrada aos algoritmos de recomendação.

Assim, esta etapa executa inúmeros processos, de forma a gerar tabelas que consigam ser utilizadas pelos algoritmos de recomendação. Vale ressaltar que a modelagem dos dados resultante geralmente não se apresenta normalizada, mas sim como sendo uma única tabela com todas as características necessárias podendo variar de algoritmo para algoritmo.

Um detalhe importante desta etapa é o agrupamento dos produtos. Esta estratégia tem dois objetivos: o primeiro é limitar a quantidade de dados que os algoritmos precisarão suportar (quanto mais grupos, mais específicos estes serão e haverá mais registros para os algoritmos processarem). O segundo objetivo é facilitar a inserção das regras de negócio ao final.

C. Algoritmos de Recomendação

Este é o núcleo do sistema de recomendação, pois contém dois ou mais algoritmos de recomendação (preferencialmente pertencentes a abordagens diferentes). As saídas destes algoritmos são, de fato, recomendações de grupos de produtos para cada cliente junto com um fator numérico, que indica quão boa é aquela recomendação. Este fator numérico servirá como base para a ponderação final.

D. Algoritmo de Ponderação:

Como existirão recomendações paralelas, será necessário fazer a escolha das N melhores recomendações. Esta escolha pode ser realizada utilizando regras triviais, como as recomendações mais frequentes ou de forma mais complexa, utilizando algoritmos de aprendizado de máquina, por exemplo.

E. Recomendação Final

Como as recomendações, até esta etapa, são baseadas em grupos de produtos, é necessário converter estes grupos em produtos (diminuir a granularidade). Neste momento, as regras de negócio da empresa são inseridas para escolher as N principais recomendações de produtos para determinado cliente dentro dos grupos de produtos previamente selecionados.

5 IMPLEMENTAÇÃO DO SISTEMA DE RECOMENDAÇÃO

A implementação das abordagens de recomendação será realizada baseada na arquitetura proposta e apresentada no capítulo anterior.

5.1 Base de dados utilizada

Para fazer a implementação e testes da arquitetura, será utilizado uma base de dados do Grupo Soma®, a qual foi gentilmente cedida. Dados pessoais ou que possam identificar os segredos industriais da empresa serão mantidos sobre sigilo, conforme requisitado pela empresa.

A base apresenta a seguinte quantidade de dados:

- Dados de mais de 3,3 milhões de compras;
- Mais de 1,15 milhão de clientes (CPF's diferentes);
- Mais de 75 mil produtos diferentes;

A base apresenta dados de produtos (grupo, subgrupo, cor, descrição, coleção, estilo, descrição, tipo, entre outros), dados de clientes (nome, CPF e algumas outras informações como data de nascimento, sexo, estado civil, mas que têm baixa confiabilidade) e dados das transações realizadas nas lojas físicas (data, loja, cliente, produto, valor e quantidade).

Estes dados estavam dentro do *Data Lake* da empresa (banco de dados MySQL) já com um tratamento básico, seguindo as regras de negócio da companhia.

5.2 Ferramentas utilizadas

Para fazer a implementação do sistema de recomendação foram utilizadas basicamente duas ferramentas, cada uma com uma finalidade específica:

- **Pentaho Data Integration® (PDI)**: Ferramenta de programação visual focada em desenvolvimento de *pipelines* de tratamento de dados. É *open source*, e possibilita trabalhar com agilidade sobre alguns milhões de registros, sendo facilmente integrada com vários bancos de dados. Foi utilizada para realizar o tratamento inicial dos dados, bem como, sincronizar a execução de cada etapa da arquitetura.
- **Python**: Linguagem de amplo uso que tem sido bastante utilizada para desenvolvimento de algoritmos de Inteligência Artificial pela facilidade de implementação e disponibilidade de inúmeras bibliotecas. Foi utilizada para

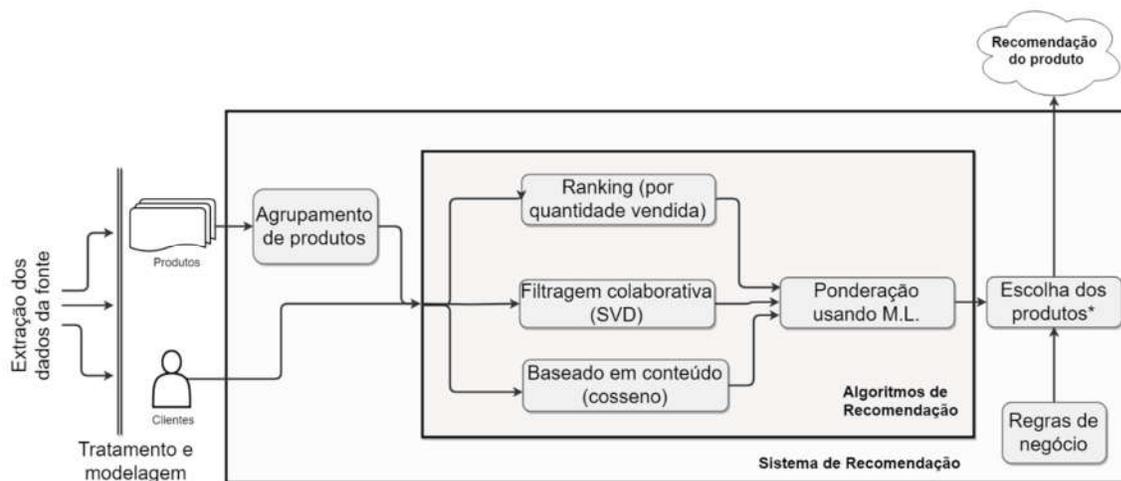
efetuar o tratamento de dados (usando a biblioteca Pandas®) e também na implementação de todos os algoritmos (utilizando as bibliotecas Numpy® e SkLearn®).

Estas ferramentas foram escolhidas pela facilidade de desenvolvimento, robustez e capacidade de escalar quando se obtém alguns milhões de recomendações sem a necessidade de grandes adaptações.

5.3 Implementação da arquitetura

O sistema de recomendação a ser implementado para o caso do Grupo Soma® utilizará as abordagens de recomendação encontradas no capítulo anterior dentro da arquitetura proposta, conforme a Figura 10.

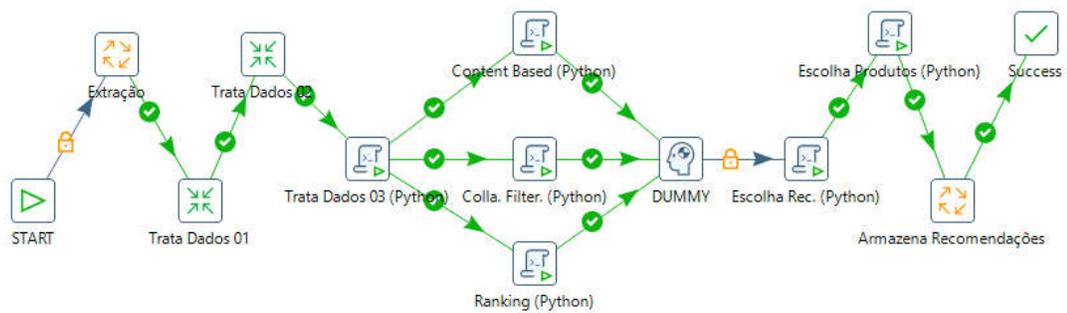
Figura 10 – Arquitetura implementada sistema de recomendação



Fonte: Autor

Este modelo foi implementado, conforme já comentado, utilizando a ferramenta PDI e Python. O modelo visual de programação do PDI permite a visualização clara das etapas da arquitetura, conforme figura 11.

Figura 11 – Implementação da arquitetura usando PDI



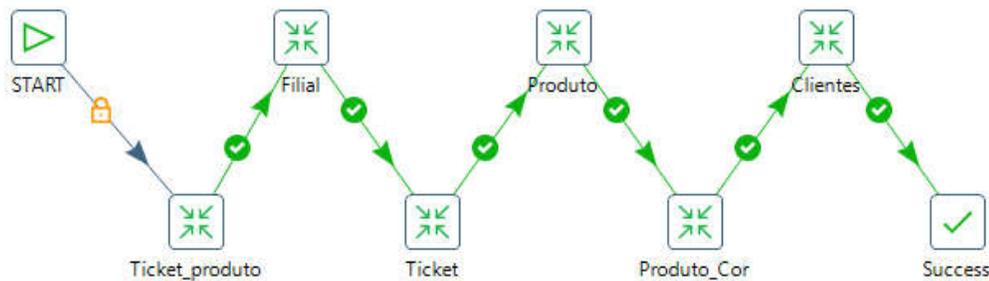
Fonte: Autor

Dentro de cada bloco mostrado na figura 13, há uma parte da arquitetura. Por isso, a seguir será feito uma descrição de como estas foram implementadas.

5.3.1 Extração dos dados

Para a etapa de extração de dados o *software* Pentaho® foi utilizado. Assim, 6 tabelas do *Data Lake* do Grupo Soma® foram copiadas para o banco de dados que será usado para gerar e armazenar as recomendações. A figura 12 mostra como esta extração foi implementada usando o PDI.

Figura 12 – Extração dos dados com o PDI



Fonte: Autor

Vale salientar que nesta etapa nenhum tratamento é realizado, ou seja, ocorre somente uma cópia fiel dos dados de origem.

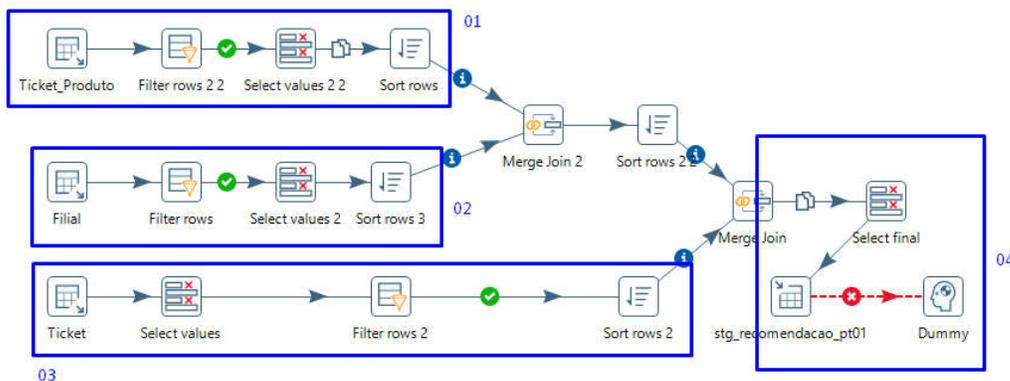
5.3.2 Tratamento dos dados

Esta etapa está dividida em 3 (três) partes, sendo que 2 (duas) foram implementadas dentro do Pentaho® e uma dentro do Python. Esta divisão foi necessária devido à complexidade deste processo, visto que:

- A. O volume de dados que está sendo utilizado é considerável (aproximadamente 20 milhões de registros antes dos tratamentos);
- B. A modelagem dos dados precisou ser completamente refeita (em relação ao que estava modelado no *data lake* da empresa). Isto ocorreu porque o *data lake* é utilizado para fins analíticos;
- C. A limpeza da base para exclusão dos registros que não serão utilizados foi elaborada usando várias regras (exclusão de lojas, exclusão de clientes, etc);
- D. Cada algoritmo de recomendação exige uma entrada de dados diferente. Assim, para esta prova de conceito foi preciso desenvolver duas modelagens completamente diferentes;

As primeiras duas etapas são responsáveis por unir os dados de várias tabelas, formando uma tabela denominada *stage*, que servirá de base para fazer um tratamento dos dados específicos para cada algoritmo dentro do Python. A Figura 13 demonstra a implementação da primeira parte.

Figura 13 – Parte 01 do tratamento dos dados



Fonte: Autor

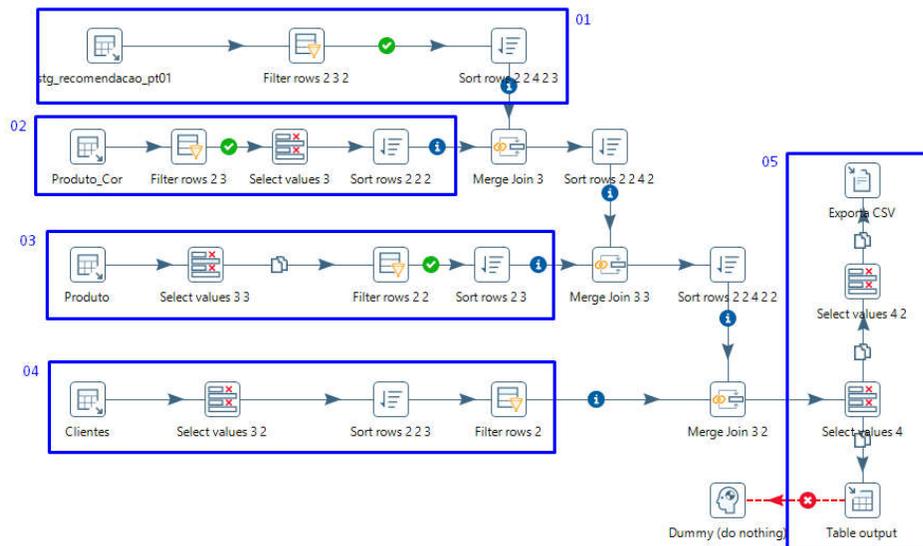
As etapas realizadas são:

- A. Tratamento da tabela de vendas por produto, para união com a tabela de lojas;

- B. Tratamento da tabela de lojas, excluindo os registros que são referentes a *e-commerce*;
- C. Tratamento da tabela de vendas para união com os dados de venda e lojas;
- D. Armazenamento do resultado dentro de uma tabela *stage* temporária no banco de dados.

A segunda parte do tratamento dos dados está implementada conforme a Figura 14.

Figura 14 – Parte 02 do tratamento dos dados



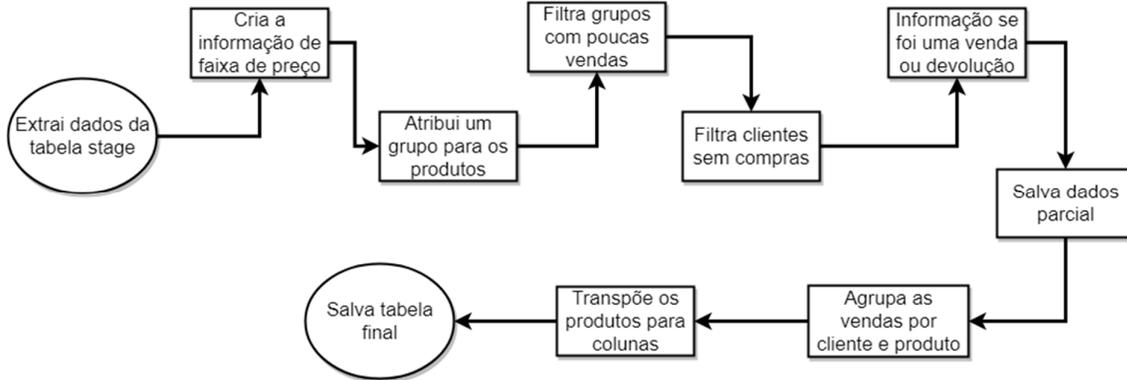
Fonte: Autor

As etapas realizadas são:

- A. Leitura da tabela de *stage* gerada na parte 01;
- B. Tratamento da tabela de cores de produto, excluindo registros inválidos;
- C. Tratamento da tabela de produtos, excluindo registros inválidos;
- D. Armazenamento do resultado dentro de uma tabela *stage* no banco de dados e em um arquivo texto, para facilitar a leitura pelo Python.

A terceira parte é implementada em Python e é responsável por criar as tabelas que serão utilizadas pelos algoritmos de recomendação. A seguir (Figura 15) é apresentado um fluxograma com as principais etapas do tratamento dos dados.

Figura 15 – Parte 03 do tratamento dos dados



Fonte: Autor

O filtro de produtos com poucas vendas (abaixo de 30 vendas no período de 6 meses) é importante para excluir recomendações de produtos que estão fora ou saindo de linha ou produtos. Um detalhe importante a salientar é que o grupo é formado pela composição da: Marca, Grupo, Subgrupo, Linha e Faixa de Preço. Assim, um grupo de produto seria “2_VESTIDOS_CURTO_TECIDO_100-200”.

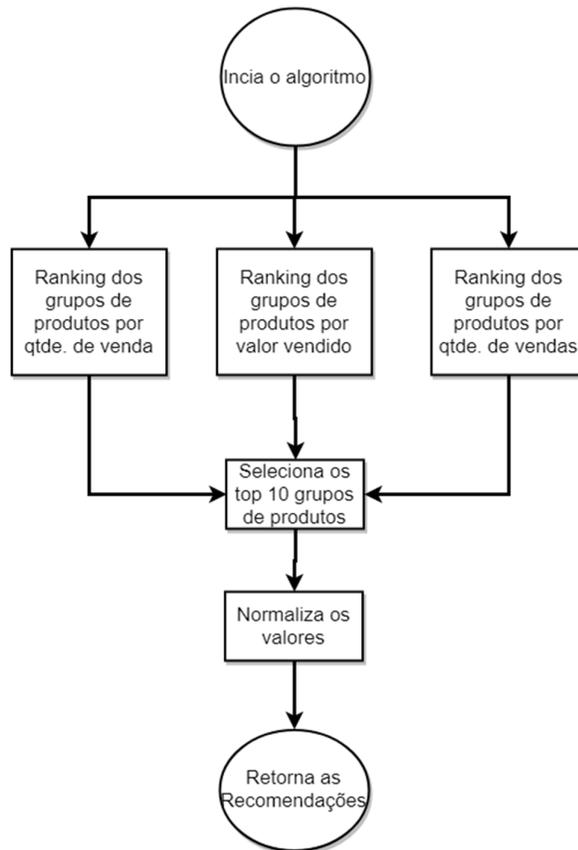
5.3.3 Algoritmo de Ranqueamento

O algoritmo de Ranqueamento foi implementado segundo 3 possíveis regras:

- A. Maior quantidade de peças vendidas no período de 6 meses;
- B. Maior valor vendido, ou seja, maior receita gerada, segundo a fórmula preço de venda x quantidade vendida – descontos;
- C. Maior quantidade de transações de venda, ou seja, maior número de compras que contém aquele produto;

A seguir (Figura 16) é apresentado um fluxograma com as principais etapas do algoritmo.

Figura 16 – Algoritmo de rankeamento



Fonte: Autor

O algoritmo inicialmente ordena os grupos de produtos segundo o critério escolhido e, após isso, seleciona os 10 grupo de produtos com maiores valores. Os valores de venda (ou quantidade de vendas ou quantidade de peças vendidas) são normalizados de forma que os números retornados estejam dentro de um intervalo de 0 a 1 (este valor será usado pelo algoritmo de escolha final para ponderar quão boa é uma recomendação em relação a outra do mesmo algoritmo). Estes dados são retornados na forma de um *DataFrame* (equivalente a uma tabela).

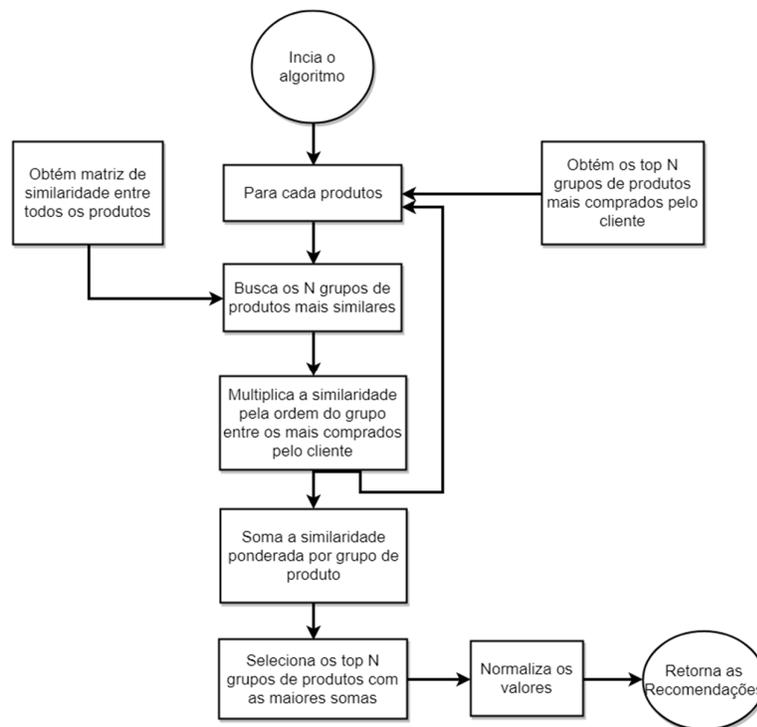
Para este estudo de caso, as recomendações foram baseadas na maior quantidade de vendas. O código com a implementação deste algoritmo se encontra no APÊNDICE B – Algoritmo de Rankeamento.

5.3.4 Algoritmo Baseado em Conteúdo

O algoritmo baseado em conteúdo busca características dos produtos para que, com base nas compras passadas de cada cliente, possa localizar produtos similares. As características escolhidas para que seja possível comparar os produtos são, Marca, Grupo, Subgrupo, Linha e Faixa de Preço.

Com estas características é realizado um cálculo de similaridade, usando o cosseno, entre todos os produtos. Os detalhes podem ser vistos no fluxograma abaixo (Figura 17).

Figura 17 – Algoritmo de recomendação baseada em conteúdo



Fonte: Autor

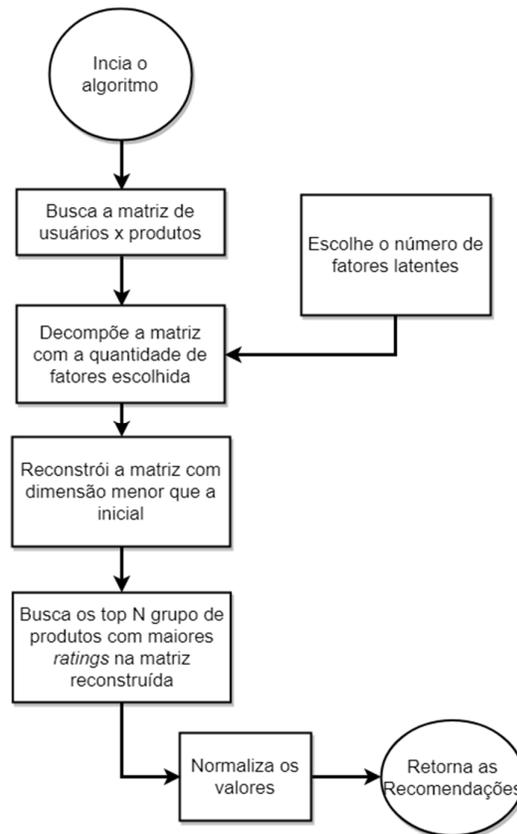
A recomendação é efetuada somando, para cada grupo de produto, a similaridade ponderada pela ordem do grupo de produtos dentro dos grupos mais comprados (ex: grupo mais comprado pelo cliente tem ordem 5, segundo mais comprado tem ordem 4, e assim por diante). Desta forma, um grupo de produto que é similar ao segundo e ao terceiro grupo de produto mais comprado pelo cliente poderá ser recomendado como o mais provável a ser adquirido pelo cliente.

Ao final, os valores resultantes das somas são normalizados entre 0 e 1 e retornados na forma de um *DataFrame*. O código com a implementação deste algoritmo está no APÊNDICE C – Algoritmo baseado em similaridades de produto.

5.3.5 Algoritmo de Filtragem Colaborativa

O algoritmo de filtragem colaborativa é o mais complexo entre todos. Ele pode ser implementado de diversas formas, como foi exposto no Capítulo 3. Para este estudo de caso, devido ao tamanho e variedade dos dados, foi escolhido a implementação baseada em fatoração de matriz usando o algoritmo SVD. Maiores detalhes podem ser vistos no fluxograma abaixo (Figura 18).

Figura 18 – Algoritmo de filtragem colaborativa com SVD



Fonte: Autor

O algoritmo utiliza a matriz de Clientes x Produtos, esta tendo uma forma similar a Figura 19.

Figura 19 – Exemplo de matriz Usuários x Produtos

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	0
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	0

Fonte: [40]

Os zeros, neste exemplo, representam itens que ainda não foram consumidos (ou avaliados) pelos usuários. O objetivo é que, com a decomposição e reconstrução da matriz, apareçam valores relevantes (diferentes de 0) nestas células que inicialmente são 0. Assim, quando maior for este valor, maior é a tendência de o consumidor aprovar a recomendação.

Para o estudo de caso, as avaliações dos produtos são obtidas implicitamente através da compra e devolução do produto. Por exemplo, um cliente que comprou duas vezes o mesmo grupo de produto receberá um *rating* de 2. Se este mesmo cliente comprou outro grupo de produto e após devolveu este produto, ao final ele terá um *rating* -1 para este grupo de produto.

5.3.5.1 Algoritmo de decomposição de matriz SVD

O algoritmo de decomposição de matriz SVD recebe como parâmetro o número de fatores latentes (fatores relevantes dos produtos). Para este caso, foram feitos alguns testes simples de acurácia baseada na métrica de RMSE e foi escolhido o número de 80.

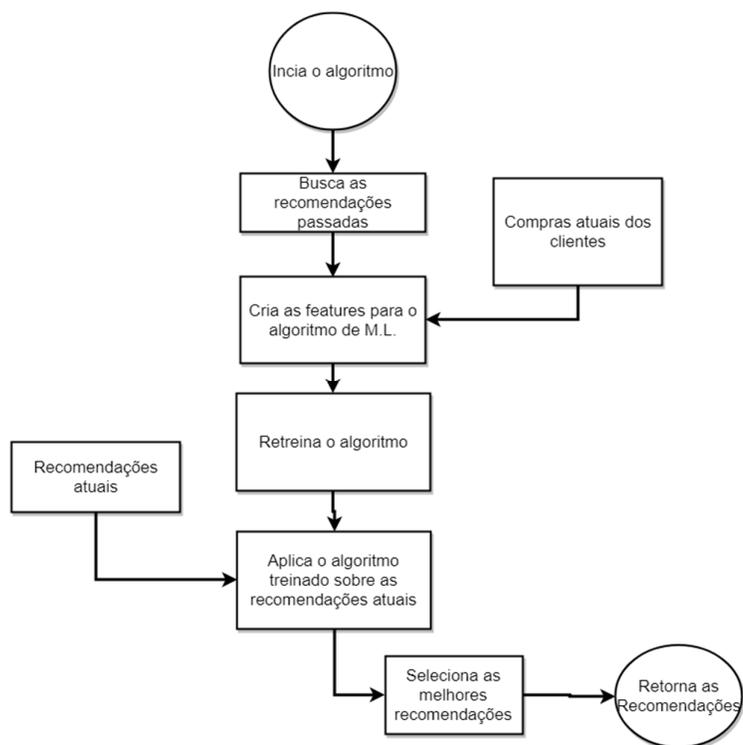
A implementação deste algoritmo foi realizada utilizando a biblioteca de computação científica SciPy (mais especificamente o seu módulo de Álgebra Linear).

O código com a implementação completa deste algoritmo está no APÊNDICE D – Algoritmo de filtragem colaborativa.

5.3.6 Algoritmo de escolha das recomendações

O algoritmo de escolha das recomendações é implementado utilizando aprendizado de máquina, de forma que esta preveja se a recomendação resultará em uma compra, nenhuma ação ou uma compra com devolução. Os detalhes da implementação são explicados no fluxograma abaixo (Figura 20).

Figura 20 – Algoritmo de escolha das recomendações



Fonte: Autor

A seguir são apresentados alguns detalhes importantes:

- A. Para realizar o treinamento do algoritmo, são utilizadas recomendações comparando-as com o que cliente comprou na sequência. Assim, a previsão que o algoritmo de aprendizado de máquina deve fazer é qual a ação que o cliente terá sobre aquele grupo de produtos: compra (melhor cenário); nada a fazer (cenário neutro) ou compra e devolução (pior cenário);

- B. Para fazer a escolha das recomendações, o algoritmo treinado é aplicado sobre as recomendações geradas pelos algoritmos de recomendação, gerando uma provável ação do cliente sobre aquele produto com um grau de certeza;
- C. Para fazer a escolha das recomendações finais, é buscado, para cada cliente, os grupos de produtos com mais alta probabilidade de compra e, caso não exista, com maiores probabilidades de serem neutros;
- D. As entradas (*features*) que o algoritmo de aprendizado de máquina recebe são a recomendação, o grau de confiança desta recomendação (número entre 0 e 1 gerado pelos algoritmos de recomendação) e o estado daquele grupo de produto no passado (se ele já foi comprado ou devolvido pelo cliente);
- E. Algumas *features* precisam ser criadas no modelo de *dummy*, ou seja, para cada classe da entrada (por exemplo, o tipo de algoritmo, Ranqueamento, Baseada em Conteúdo ou Filtragem Colaborativa), é necessário criar uma coluna contendo 1 caso aquele exemplo pertença aquela classe, 0 caso contrário.

5.3.6.1 Algoritmo de *Machine Learning*

Para auxiliar na escolha da recomendação, é utilizado um algoritmo de aprendizado de máquina chamado *Gradiente Boosting* baseado em árvores de decisão. O ajuste de parâmetros é obtido utilizando dados de teste e treinamento. A implementação foi realizada baseada na biblioteca Sklearn. O código com a implementação do algoritmo de aprendizado de máquina está no APÊNDICE E – Algoritmo de Escolha final.

5.3.7 Escolha dos produtos com regras de negócio

A escolha de produtos está submetida a uma regra parametrizável. Esta regra pode ser algo extremamente simples, como selecionar apenas produtos da coleção atual ordenados pelos maiores valores de vendas, até regras mais complexas que relacionem o estoque dos produtos, preços dos produtos e margens de venda. Em um ambiente real, estas regras deveriam estar disponíveis para os analistas de negócio da empresa alterarem e parametrizarem conforme necessário.

Nesta prova de conceito implementada, foram desenvolvidas 2 regras (maior quantidade de vendas e maior margem de lucro) e sendo definidas diretamente no

código da aplicação. O código está no APÊNDICE F – Recomendação de produtos com regras de negócio.

6 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Tão importante quanto o desenvolvimento do sistema é a mensuração dos seus resultados. Para esse objetivo, foram escolhidas três métricas quantitativas para verificar a acurácia do sistema. Além disso, será feita uma avaliação se a arquitetura proposta e o sistema desenvolvido conseguiu cumprir todos os requisitos iniciais. Ao final, será realizada uma avaliação dos resultados dentro de uma perspectiva, bem como, sobre uma perspectiva de negócios.

6.1 Procedimento de teste

Para a realização dos testes de acuracidade o sistema usou dados passados permitindo realizar as etapas de treinamento e teste. Assim, foram usados dados até 31 de dezembro de 2017 para realizar o treinamento dos algoritmos de recomendação. Após, foram geradas aproximadamente 5 milhões de recomendações de grupos de produtos para 246 mil clientes diferentes (CPF's distintos) que fizeram compra entre janeiro e março de 2018.

Destes 246 mil clientes, 46 mil foram selecionados para serem utilizados como base de teste. O restante, 200 mil clientes, foram usados como base para o treinamento do algoritmo de escolha das recomendações. Após o treinamento dos algoritmos, estes foram aplicados para os 46 mil clientes, gerando 54 mil recomendações de grupos de produtos.

Estas milhares de recomendações foram comparadas com as compras destes clientes dentro dos meses de janeiro a março de 2018. A partir da comparação entre as compras reais e as recomendadas foram calculadas as métricas do sistema de recomendação.

As limitações deste procedimento de teste são:

- A. O sistema foi testado sobre dados de compra de clientes que não tiveram influência de um sistema de recomendação. Porém, o simples fato de existir recomendações podem influenciar o comportamento de compra do consumidor, aumentando a acurácia do sistema. A única maneira de transpor esta limitação é implementado o sistema junto a um grupo de clientes e mensurar as compras dos clientes que tiveram influência do sistema de recomendação com aqueles que não tiveram influência, como um teste A/B;

- B. O teste leva em conta grupo de produtos, não produtos. Isto ocorre porque o sistema, ao final, recomenda um grupo de produtos que será transformado em recomendações de produtos conforme o que a empresa quer otimizar. Isto não significa, necessariamente, que a empresa queira otimizar a quantidade de vendas ou de recomendações que terão uma maior acurácia;
- C. A janela de tempo da avaliação da compra é de 3 meses, porém o sistema não foi retreinado com uma periodicidade. Por exemplo, a cada semana, os algoritmos poderiam ser retreinados para capturar os novos dados, atualizando e alterando as recomendações. Isto limita a acuracidade do sistema, uma vez que a implementação deste método de treinamento seria muito custosa;
- D. Os algoritmos foram ajustados para otimizar os acertos dentro desta janela de tempo, isso significa que eles podem diminuir a acurácia do modelo prevendo novos dados. Isto pode ser resolvido modificando os hiperparâmetros dos algoritmos, de forma que estes consigam ter um maior poder de generalização dos dados;
- E. As métricas não levam em consideração pontos específicos desempenho como novos produtos, novos clientes ou datas comemorativas. Contudo, estes pontos podem, em futuras melhorias, receber tratamento personalizado a fim de atingirem um melhor desempenho caso necessário.

6.2 Resultados Obtidos

A seguir serão apresentados os resultados do sistema implementado.

6.2.1 Acuracidade

A medida de acuracidade é bastante importante já que é uma premissa básica para um sistema ser, de fato, aplicável que ele seja minimamente assertivo. Assim, foram utilizados três métricas de avaliação (estas já foram discutidas no Capítulo 3).

6.2.1.1 RMSE

A medida define o percentual médio de erro. Como explicado nos capítulos anteriores, este valor foi obtido dividindo as recomendações erradas pelo total de recomendações. Assim, o resultado obtido dentro do cenário de teste foi de 98,9%.

Apesar de um percentual elevado, há uma importante ressalva: uma recomendação é tentar acertar o que um cliente irá comprar dentro de um período limitado de tempo. Esta decisão de compra é definida por muitos fatores (promoções de concorrentes, necessidade de compra daquele produto, datas especiais, influência de amigos e parentes, etc), os quais a maioria delas não podem ser mensuradas. Por isso, erros desta magnitude são esperados neste contexto. Possíveis melhorias serão discutidas no próximo capítulo.

Apenas a título de comparação, uma recomendação aleatória de grupos de produtos geraria um RMSE de mais de 99,95% (há aproximadamente 2 mil e 100 grupos de produtos possíveis de serem recomendados). Ou seja, o sistema implementado acerta aproximadamente 20 vezes a mais. Também apenas a título de comparação, durante a competição de recomendações da Netflix® em 2007, os melhores resultados de recomendação ficaram com RMSE na casa de 88% [41]. O cenário da Netflix® é muito diferente, dado a facilidade de consumo da recomendação (assistir um filme não implica em gasto monetário, a compra de um produto implica). Além disso, houve uma enorme otimização dos algoritmos implementados para o cenário específico da empresa (este sistema é considerado muito próximo ao estado da arte dos sistemas de recomendação, sendo estudado e testado por cerca de 2 anos), também as recomendações da Netflix foram apresentadas aos usuários do sistema, influenciando no positivamente na acurácia do sistema (conforme item A do subcapítulo Procedimento de Teste)

6.2.1.2 Precision

Precision mede a taxa de falsos positivos (estimativa de compra quando não há compra) dentro do total de recomendações. Como o sistema de recomendação gera apenas recomendações da classe em que é esperado uma compra pelo cliente, esta métrica é equivalente a taxa de acertos do sistema. Assim, o número obtido é de 1,04%. A análise anterior também é válida para este indicador: apesar de o número não ser tão expressivo, a especificidade que requer uma recomendação assertiva é tamanha que não era esperado um número muito grande para este indicador.

6.2.1.3 MAP

Esta medida é mais complexa de ser compreendida que as outras, pois mensura, não um resultado absoluto, mas sim uma média de quando o sistema erra na lista de recomendações. Assim, o valor obtido é de 0,008. Isto significa que, em média, ponderando os resultados pela sua ordem, acertamos 0,8%. Este valor é menor que o valor de *precision*, ou seja, há recomendações assertivas que não estão listadas como as primeiras.

6.2.2 Requisitos do sistema

No capítulo 4 foram traçados 11 requisitos que o sistema deveria cumprir. A seguir é apresentada uma avaliação para cada requisito:

- 1- Assertividade: avaliada na seção 6.2.1. O sistema consegue fazer recomendações relevantes para o cliente (será demonstrado exemplos a seguir), tendo níveis de assertividade compatíveis com o cenário. Requisito cumprido;
- 2- Regras de negócio: é possível inserir regras de negócio para priorização de produtos, otimizar estoques ou margens de vendas, por exemplo. Além disso, é possível, durante o tratamento dos dados, excluir produtos ou clientes em específico. Requisito cumprido;
- 3- Dados gerais de produtos: as 5 informações utilizadas sobre produtos são bastante gerais. Tanto a empresa em que este estudo foi realizado quanto outras de varejo físico já dispõem dessas informações em seus bancos de dados. Isso torna o sistema implementado aplicável em empresas deste ramo. Requisito cumprido;
- 4- Dados demográficos: nenhum dado demográfico dos clientes foi utilizado. Requisito cumprido;
- 5- Escalabilidade: como os algoritmos utilizados são simples, além de sempre usar grupos de produtos, é possível escalar o sistema sem maiores problemas. Requisito cumprido;
- 6- Implementação: as bibliotecas e frameworks existentes aceleram muito o desenvolvimento do sistema, porém o tempo de implementação não é o menor possível, visto que são necessários vários algoritmos para o desenvolvimento da arquitetura. Requisito parcialmente cumprido;

- 7- Estabilidade: como existem três algoritmos de recomendação e um “filtro” ao final com o objetivo de retirar recomendações não adequadas, o sistema tende a fazer recomendações, no mínimo, aceitáveis. Requisito cumprido;
- 8- *Feedbacks* negativos: são utilizados apenas *feedbacks* negativos de devolução (bastante raros). Requisito cumprido;
- 9- Recomendações explicáveis: como são acoplados algoritmos em série, para explicar uma recomendação seria necessário percorrer um caminho longo (apesar de ainda ser possível). Requisito parcialmente cumprido;
- 10- Integração: é possível integrar o sistema através de serviços que fazem a leitura das recomendações armazenadas no banco de dados. Requisito cumprido;
- 11- *Cold start*: nenhuma ação específica foi tomada em relação a recomendações para clientes que estão fazendo a primeira compra na empresa. Para clientes com poucas compras, as recomendações baseadas em similaridade de produtos tendem a funcionar melhor. Requisito parcialmente cumprido;

Dos 11 requisitos levantados, 3 foram parcialmente cumpridos e 8 deles foram totalmente cumpridos, mostrando a viabilidade implementação de um sistema de recomendação para o setor de varejo físico.

6.3 Exemplos de recomendação obtidas

Para deixar mais claro as recomendações realizadas, serão apresentados dois exemplos de recomendações. Estes exemplos foram retirados da base de teste, ou seja, foram calculados usando os métodos já discutidos. Estes exemplos estão gravados em um banco de dados (assim como todos os outros clientes de testes) e foram acessados usando um *web service* criado para esta finalidade. As informações pessoais dos clientes são omitidas, bem como alguns dados dos produtos.

6.3.1 Exemplo 01

A cliente deste primeiro exemplo comprou 7 vezes nos últimos 6 meses de 2017, sendo os seguintes grupos de produtos:

Tabela 9 - Compras realizadas pelo cliente do exemplo 1

Marca	Grupo	SubGrupo	Linha	Preço Varejo	Produtos Comprados
2	Chinelo	Chinelo	Havaiana	X54	3
2	Blusa	Manga Curta	Tricot	X69	1
2	Vestido	Curto sem MG	Tecido	X29	1
2	Macacão	Curto sem MG	Tecido	X49	1
2	Vestido	Curto sem MG	Jeans	X69	1
2	Vestido	Curto	Tecido	X29	1
2	Vestido	Especial	Malha	X39	1
2	Macacão	Curto	Malha	X98	1
2	Short	Cintura Alta	Esporte	X79	1
2	Blusa	Especial	Malha	X49	1

Fonte: Autor

As informações de preços foram ocultadas para manter o sigilo da fonte de dados. A coluna “prod_comprados” denomina a quantidade de produtos comprados daquele grupo. Há uma tendência para a compra de vestidos, especialmente os do subgrupo curto. A seguir são apresentadas as recomendações de grupos de produtos de cada algoritmo (Tabelas 22 a 25), inclusive os grupos de produtos que foram selecionados pelo algoritmo de escolha final.

Tabela 10 - Recomendações geradas pelo algoritmo de ranqueamento

Marca	Grupo	SubGrupo	Linha	Faixa de Preço
2	Vestido	Longo	Tecido	400-800
2	Vestido	Curto	Tecido	200-400
2	Macacão	Curto	Tecido	200-400
2	Blusa	Regata	Tecido	100-200
1	Vestido	Curto	Seda	800-1600
1	Vestido	Curto	Seda	400-800
1	Top Basico	Blusa	Malha	100-200
1	Top	Blusa	Seda	400-800
1	Over Tops	Jaqueta	Couro	2400-4800
1	Camisa	Camisa	Seda	400-800

Fonte: Autor

Tabela 11 - Recomendações geradas pelo algoritmo *content based*

Marca	Grupo	SubGrupo	Linha	Faixa de Preço
2	Vestido	Curto	Tricot	200-400
2	Vestido	Curto	Tecido	200-400
2	Vestido	Curto	Tecido	100-200
2	Vestido	Curto	Malha	200-400
2	Vestido	Curto	Artesanal	200-400
2	Short	Curto	Tecido	200-400
2	Macacão	Curto	Tecido	200-400
2	Chinelo	Chinelo	Sapato	100-200
2	Chinelo	Chinelo	Havaiana	0-100
1	Vestido	Curto	A+	200-400

Fonte: Autor

Tabela 12 - Recomendações geradas pelo algoritmo de filtragem colaborativa

Marca	Grupo	SubGrupo	Linha	Faixa de Preço
2	Vestido	Curto	Tecido	400-800
2	Sapato	Tenis	Sapato	200-400
2	Macacão	Curto	Tecido	100-200
2	Macacão	Curto	Malha	100-200
1	Chinelo	Chinelo	Havaiana	0-100
1	Bolsa	Mochila	Cobranding	200-400
1	Blusa	Top	Malha	100-200
1	Top	Regata	Tecido	100-200
1	Over Tops	Cropped	Malha	100-200
1	Agasalho	Jaqueta	Cobranding	200-400

Fonte: Autor

Tabela 13 - Recomendações finais escolhidas

Marca	Grupo	SubGrupo	Linha	Faixa de preço
2	Vestido	Curto	Tecido	200-400
2	Blusa	Regata	Tecido	100-200
1	Vestido	Curto	Seda	400-800

Fonte: Autor

Os grupos de produtos escolhidos pelo algoritmo final estavam presentes nos algoritmos baseado em conteúdo, filtragem colaborativa e ranqueamento (a presença de uma recomendação final de cada algoritmo intermediário é apenas uma coincidência, não tendo nenhuma regra que priorize este tipo de recomendação).

Neste caso, nenhum dos grupos de produtos recomendados foi comprado pelo cliente (apesar das recomendações estarem de acordo com o esperado segundo as compras passadas do cliente). Com estes grupos, foram geradas as seguintes recomendações de produtos, conforme figura 28.

Figura 21 – Recomendações finais de produtos

```

{
  "cliente": "4[REDACTED]1",
  "produtos": [
    {
      "desc_produto": "VESTIDO BOUQUET DE TUCANO E",
      "grupo": "VESTIDOS",
      "subgrupo": "CURTO",
      "colecacao": "AV18",
      "linha": "TECIDO",
      "estilista": "CRIS",
      "preco_varejo_original": [REDACTED]
    },
    {
      "desc_produto": "VESTIDO BOTOES TODA FLORIDA",
      "grupo": "VESTIDOS",
      "subgrupo": "CURTO",
      "colecacao": "INV18",
      "linha": "TECIDO",
      "estilista": "THALITA",
      "preco_varejo_original": [REDACTED]
    },
    {
      "desc_produto": "VESTIDO CURTO LENCO ABACAXINA",
      "grupo": "VESTIDOS",
      "subgrupo": "CURTO",
      "colecacao": "INV18",
      "linha": "TECIDO",
      "estilista": "THALITA",
      "preco_varejo_original": [REDACTED]
    }
  ]
}

```

Fonte: Autor

As recomendações de produtos foram efetuadas basicamente no grupo “Vestidos”, já que este é um padrão de compra recorrente do cliente. As compras deste cliente, no período de teste, seguiram o padrão esperado dentro do grupo de “Vestidos”.

Tabela 14 - Compras do cliente do exemplo 01

Marca	Grupo	SubGrupo	Linha	Faixa de preço
2	Vestido	Curto sem MG	Jeans	200-400
2	Vestido	Especial	Malha	200-400

Fonte: Autor

As compras são similares com as recomendações, tendo comprado vestidos, sendo um curto (como eram as recomendações). Este é um exemplo que as recomendações estão adequadas com o cenário, porém como a dificuldade de acertar exatamente o produto que o cliente irá comprar é muito grande e há uma série de variáveis que são impossíveis de serem mapeadas, o indicador RMSE para este cliente em específico foi de 100%.

6.3.2 Exemplo 02

Este segundo exemplo é de um cliente com as seguintes compras, conforme Figura 28.

Tabela 15 - Compras realizadas pelo cliente do exemplo 2

Marca	Grupo	SubGrupo	Linha	Preço Varejo	Produtos Comprados
1	Top Basico	Blusa	Malha	X98	3
1	Vestido	Vestido	Seda	X98	1
1	Camisa	Camisa	Seda	X98	1
1	Top Neutro	Blusa	Malha	X68	1
1	Bottom	Bermusa	Sarja	X68	1
1	Top Fun	T-Shirt	Malha	X98	1
2	Blusa	Body	Malha	X49	1

Fonte: Autor

Com estas compras, foram realizadas as seguintes recomendações para este cliente (Tabelas 29 a 32).

Tabela 16 - Recomendações geradas pelo algoritmo de ranqueamento

Marca	Grupo	SubGrupo	Linha	Faixa de Preço
2	Vestido	Longo	Tecido	400-800
2	Vestido	Curto	Tecido	200-400
2	Macacão	Curto	Tecido	200-400
2	Blusa	Regata	Tecido	100-200
1	Vestido	Curto	Seda	800-1600
1	Vestido	Curto	Seda	400-800
1	Top Basico	Blusa	Malha	100-200
1	Top	Blusa	Seda	400-800
1	Over Tops	Jaqueta	Couro	2400-4800
1	Camisa	Camisa	Seda	400-800

Fonte: Autor

Tabela 17 - Recomendações geradas pelo algoritmo baseado em conteúdo

Marca	Grupo	SubGrupo	Linha	Faixa de Preço
8	Top Elaborado	Blusa	Malha	100-200
4	Top Basico	Top	Malha	100-200
2	Blusa	Tunica	Malha	100-200
2	Blusa	Top	Malha	100-200
2	Blusa	Manga Curta	Malha	100-200
2	Blusa	Body	Malha	100-200
1	Top Neutro	Blusa	Malha	100-200
1	Top Fun	Blusa	Malha	100-200
1	Top Basico	Blusa	Malha	200-400
1	Top Basico	Blusa	Malha	100-200

Fonte: Autor

Tabela 18 - Recomendações geradas pelo algoritmo de filtragem colaborativa

Marca	Grupo	SubGrupo	Linha	Faixa de Preço
2	Blusa	Body	Malha	100-200
1	Top Fun	Blusa	Malha	100-200
1	Top Basico	Blusa	Malha	200-400
1	Top Basico	Blusa	Malha	100-200
1	Top	Blusa	Resort	200-400
1	Lenço	Lenço	Acessorios	200-400
1	Cintos	Couro	Acessorios	200-400
1	Calça	Skinny	Jeans	400-800
1	Calça	Calça	Tecido Plano	400-800
1	Bottom	Calça	Malha	400-800

Fonte: Autor

Tabela 19 - Recomendações finais escolhidas

Marca	Grupo	SubGrupo	Linha	Faixa de preço
1	Top Neutro	Blusa	Malha	100-200
1	Top Basico	Blusa	Malha	100-200

Fonte: Autor

Este cliente possui uma tendência a comprar produtos do grupo “Top”, e suas variações. Estas recomendações finais podem ser encontradas nos algoritmos de filtragem colaborativa e baseada em conteúdo, respectivamente. Ambas as recomendações de grupos de produtos foram compradas por este cliente no período

de testes, ou seja, estas duas recomendações foram assertivas. Os produtos recomendados para este cliente podem ser vistos na Tabela 33.

Figura 22 – Recomendações finais de produtos

```

{
  "cliente": "03335",
  "produtos": [
    {
      "desc_produto": "TEE DRAGÃO GRAVATARIA AZUL",
      "grupo": "TOPBASICO",
      "subgrupo": "BLUSA",
      "colecacao": "VER18",
      "linha": "MALHA",
      "estilista": "CLARICE",
      "preco_varejo_original": 100
    },
    {
      "desc_produto": "TEE ORIGAMI GEO VERMELHO",
      "grupo": "TOPBASICO",
      "subgrupo": "BLUSA",
      "colecacao": "VER18",
      "linha": "MALHA",
      "estilista": "CLARICE",
      "preco_varejo_original": 100
    },
    {
      "desc_produto": "TEE FLOR BICOLOR CRU",
      "grupo": "TOPBASICO",
      "subgrupo": "BLUSA",
      "colecacao": "VER18",
      "linha": "MALHA",
      "estilista": "CLARICE",
      "preco_varejo_original": 100
    }
  ]
}

```

Fonte: Autor

As compras realizadas por este cliente no período de teste estão na Tabela 34.

Tabela 20 - Compras do cliente do exemplo 02

Marca	Grupo	SubGrupo	Linha	Faixa de preço
1	Top Basico	Blusa	Malha	100-200
1	Top Neutro	Blusa	Malha	100-200
1	Vestido	Vestido	Seda	200-400
1	Camisa	Camisa	Seda	100-200
1	Top Fun	T-Shirt	Malha	100-200

Fonte: Autor

É possível notar que o algoritmo conseguiu capturar a tendência central do cliente, ou seja, compra de produtos do grupo “Top”. Contudo, se percebe que não foi possível diversificar as recomendações para os outros grupos, como “Vestidos” ou “Camisas”, que também foram compras realizadas por este cliente.

6.4 Avaliação dos Resultados

Com o intuito de realizar uma análise mais prática dos resultados do sistema de recomendação, se faz necessário discutir a perspectiva financeira. O Grupo Soma fatura, em números absolutos, R\$1,3 bilhão anualmente [42]. Parte significativa desta receita vem de vendas no atacado e *e-commerce* (áreas em que o sistema de recomendação não é aplicável), porém ainda há uma quantidade significativa deste valor em jogo (valores reais não podem ser divulgados a pedido da empresa).

Em um cenário fictício onde metade do faturamento da empresa vem do varejo físico, 1% de acerto obtido nos testes do sistema proposto significa mais de R\$7 milhões de faturamento que seriam subsidiados por um sistema de recomendação.

De outro modo, dentro da questão de negócios, a empresa poderia, utilizando as regras de negócio a seu favor:

- Otimizar estoques que estão parados, recomendando estes produtos e fomentando a sua venda;
- Impulsionar a venda de produtos com alta margem de lucro;
- Melhorar a performance de vendas no geral;
- Testar novas coleções com usuários interessados nos produtos;

Entretanto, para que estes resultados fossem possíveis, seria necessário aprimorar e refinar o sistema de recomendação. Além disso, seria necessário um segundo sistema que apresente as recomendações de forma clara para o usuário (aqui pode ser tanto um(a) vendedor(a) de loja como o cliente final) bem como realizar treinamento da equipe de vendas para incentivar a vendas destas recomendações (já que estas estão perfeitamente alinhadas com os objetivos da empresa) e também aperfeiçoar e maximizar a coleta de dados dos clientes (quanto mais dados demográficos melhor).

7 PROPOSTAS DE MELHORIA

Este capítulo do trabalho é dedicado a explicar as possíveis melhorias do sistema de recomendação, para que ele obtenha uma maior acurácia e consiga se adequar melhor aos requisitos traçados.

7.1 Melhora da acurácia das recomendações

A acurácia obtida, do ponto de vista financeiro, pode ser considerada satisfatória. Entretanto, pode ser otimizada levando-se em conta os pontos listados a seguir.

7.1.1 Otimização dos algoritmos de recomendação

Os parâmetros dos algoritmos foram escolhidos com base em testes simples. Por exemplo, para o teste das recomendações utilizou-se um conjunto de treinamento e de teste divididos de forma arbitrária. Uma opção mais robusta e geralmente utilizada em sistemas de aprendizado de máquina é a técnica de *cross-validation*. Esta técnica divide um conjunto de dados em K conjuntos objetivando avaliar, de maneira mais adequada, a capacidade de generalização de um sistema.

Em relação ao algoritmo de escolha, este não utiliza qualquer dado do cliente ou dados como período do ano, a loja, entre outros. A utilização desses dados pode aumentar significativamente a acurácia do sistema desenvolvido.

7.1.2 Diminuição do tamanho dos grupos de produtos

Os grupos de produtos estão considerando apenas 5 parâmetros. Dessa forma, pode haver muitos produtos em cada grupo e, escolher entre eles, ainda pode ser difícil, gerando erros na recomendação final. Uma abordagem com grupos mais detalhados deve produzir uma precisão maior na recomendação final de produtos, independentemente da regra de negócio escolhida.

7.1.3 Melhora do algoritmo de recomendação não personalizada

O algoritmo de recomendação não personalizado está fazendo uma recomendação extremamente genérica, principalmente neste cenário que há muitos grupos de produtos. Neste cenário, uma recomendação semi-personalizada, buscando dados simples como loja em que o cliente se encontra, dia da semana e

estação do ano poderiam limitar o universo de produtos possíveis a serem recomendados e aumentar o desempenho do algoritmo na assertividade das recomendações.

7.2 Outras melhorias

Segue possível melhoria não relacionadas com a acurácia.

7.2.1 Implementação do treinamento *online* dos algoritmos

O sistema implementado não tem a função de atualizar o algoritmo baseado apenas na variação de novas informações, ou seja, cada vez que novos dados são acrescentados ao banco de dados, o sistema precisa ser novamente treinado. Isto significa que é impossível implementar uma maneira de que o algoritmo se adapte muito rapidamente às vendas. Uma etapa de treinamento *online* utilizando apenas os dados adicionais seria interessante e proveitoso.

8 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo o desenvolvimento sistema de recomendação aplicado ao varejo. Para tal, foi proposta uma arquitetura baseada em *ensemble learning* que se mostrou bastante adequada no cenário de teste. Os resultados percentuais obtidos estão longe do estado da arte, mesmo assim, os resultados financeiros e de negócio podem ser expressivos e demonstram potencial de utilização do sistema em um ambiente real. Apesar disto, para que o sistema esteja pronto para ser disponibilizado considerando centenas de milhares de clientes, é necessário que melhorias e modificações, algumas descritas no capítulo anterior, sejam realizadas.

Em relação aos objetivos, conseguiu-se cumprir a proposta que era provar a viabilidade técnica de aplicar sistema de recomendação no cenário de varejo físico, com uma prova de conceito usando dados reais. Vale ressaltar que as avaliações foram efetuadas sobre um cenário com algumas limitações e não podem ser expandidos para todos os casos de varejo físico.

Em relação a tecnologia, o grau de maturidade das empresas brasileiras justifica o porquê não é comum sistemas de recomendação aplicados ao varejo físico. As empresas ainda estão em uma maturidade de dados anterior a de prescrição de ações (onde se encaixam os sistemas de recomendação), buscando apenas a visualização dos dados com o planejamento e tomada de ação sendo realizado por pessoas. O processo de maturação das empresas está acontecendo com certa velocidade. Todavia, ainda terá que se esperar alguns anos até se ver esta, e outras tecnologias de dados e de Inteligência Artificial, serem aplicadas no dia a dia das empresas nacionais.

Em relação a escolha correta das ferramentas de trabalho, esta mostrou-se muito importantes para o bom desenvolvimento do trabalho. A alternância entre Python e Pentaho, cada um fazendo o que foi projetado para fazer, foi fundamental para chegar ao final com resultados concretos. Isto também é valido para a utilização de bibliotecas do Python, que implementam, de forma muito eficiente, alguns dos algoritmos que foram utilizados neste trabalho. Além disso, todo o trabalho foi realizado utilizando computação na nuvem, sendo isto o que permitiu o desenvolvimento do sistema com o volume de dados utilizado (milhões de registros).

De modo geral, este trabalho mostrou-se bastante desafiador devido principalmente a quantidade de dados com a qual foi trabalhada, bem como, para o entendimento das diversas abordagens de sistemas de recomendação. Este fator foi agravado devido as limitações impostas pelo cenário de trabalho (varejo físico). A grande maioria dos casos encontrados na literatura trata exclusivamente do mundo digital.

BIBLIOGRAFIA

1. SOUZA, M. D. A influência da Internet e suas ferramentas no ambiente corporativo. Administradores, 2011. Disponível em: <<http://www.administradores.com.br/artigos/marketing/a-influencia-da-internet-esuas-ferramentas-no-ambiente-corporativo/56354/>>. Acesso em: 18 nov. 2018.
2. YAM, P. Dados públicos incentivam negócios. Scientific American Brasil. Disponível em: <http://www2.uol.com.br/sciam/noticias/dados_publicos_para_incentivar_negocios.html>. Acesso em: 18 nov. 2018.
3. GREGO, M. Conteúdo digital dobra a cada dois anos no mundo. Exame, 2014. Disponível em: <<https://exame.abril.com.br/tecnologia/conteudo-digital-dobra-a-cada-dois-anos-no-mundo/>>. Acesso em: 01 out. 2018.
4. THE ECONOMIST. The world's most valuable resource is no longer oil, but data. The Economist, 2017. Disponível em: <<https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data>>. Acesso em: 05 nov. 2018.
5. OUTLIER AI. The Future of Business Intelligence: Business Analysis Automation. Medium, 2017. Disponível em: <<https://medium.com/@outlierai/the-future-of-business-intelligence-business-analysis-automation-4f60808e7256>>. Acesso em: 18 nov. 2018.
6. ORNELAS, R. CIO. Aliada ao BI a Inteligência Artificial beneficiará a gestão do negócio, 2017. Disponível em: <<http://cio.com.br/opiniaio/2017/04/06/aliada-ao-bi-a-inteligencia-artificial-beneficiara-a-gestao-do-negocio/>>. Acesso em: 18 nov. 2018.
7. SIGMOIDAL. Recommendation Systems – How Companies are Making Money. Sigmoidal. Disponível em: <<https://sigmoidal.io/recommender-systems-recommendation-engine/>>. Acesso em: 18 nov. 2018.
8. LEVY, M.; WEITZ, B. A. Retailing management, 5º ed. [S.l.]: Boston: McGraw-Hill Irwin, 2004.
9. PELIZON, L. <https://dev.meioemensagem.com.br/home/proxima/how-to/2017/08/18/varejo-fisico-e-online-precisam-andar-de-maos-dadas.html>. Varejo físico e online precisam andar de mãos dadas, 2017. Disponível em:

<<https://dev.meioemensagem.com.br/home/proxima/how-to/2017/08/18/varejo-fisico-e-online-precisam-andar-de-maos-dadas.html>>. Acesso em: 18 nov. 2018.

10. SBVC. O papel do varejo na economia brasileira. [S.l.]. 2018.

11. TREVIZAN, K. Faturamento do comércio eletrônico cresce 7,5% em 2017, com aumento no número de pedidos. G1 Globo, 2018. Disponível em: <<https://g1.globo.com/economia/noticia/faturamento-do-comercio-eletronico-cresce-75-em-2018-com-aumento-no-numero-de-pedidos.ghtml>>.

12. MEIR, J. O futuro do varejo físico em um mundo pós-digital. Portal Novarejo, 2018. Disponível em: <<https://portalnovarejo.com.br/2018/03/o-futuro-varejo-fisico-mundo-pos-digital/>>.

13. EBIT. 36° Webshoppers. [S.l.]. 2016.

14. MAZETO, T. 7 motivos que levam ao crescimento do mercado eletrônico no Brasil. Ecommerce Brasil, 2018. Disponível em: <<https://www.ecommercebrasil.com.br/artigos/crescimento-do-mercado-eletronico/>>.

15. ESTADÃO CONTEÚDO. Vendas em lojas físicas caem no país e varejo online cresce 36,2%. Exame, 2017. Disponível em: <<https://exame.abril.com.br/economia/vendas-em-lojas-fisicas-caem-no-pais-e-varejo-online-cresce-362/>>. Acesso em: 18 dez. 2018.

16. MCKINSEY. How retailers can keep up with consumers. McKinsey.com, 2013. Disponível em: <<https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers>>.

17. SUN, C. E. A. Big data based retail recommender system of non E-commerce. Fifth International Conference on Computing, Communications and Networking Technologies, p. 1-7, jul. 2014.

18. JALLOULIA, ET AL. Designing Recommender System: Conceptual Framework and Practical Implementation. Procedia Computer Science, p. 1701-1710, 2017.

19. BHATNAGAR, V. Collaborative Filtering Using Data Mining and Analysis. [S.l.]: IGI Global, 2017.

20. FORTUNE. Amazon's recommendation secret. Fortune, 2012. Disponível em: <<http://fortune.com/2012/07/30/amazons-recommendation-secret/>>.

21. KHATWANI, S. & C. M. B. Building Personalized and Non Personalized recommendation systems. International Conference on Automatic Control and Dynamic Optimization Techniques, p. 623-628, set. 2016.
22. LU ET AL. Recommender system application developments: A survey. Decision Support Systems, p. 12-32, 2015.
23. HIRALALL, M. Recommender systems for e-shops. Business Mathematics and Informatics paper, 2011.
24. PORIYA, E. A. Non-Personalized Recommender Systems and User-based Collaborative Recommender Systems. International Journal of Applied Information Systems, 2014.
25. RICCI, A. Recommender Systems. [S.I.]: Springer, 2010.
26. SU, X. Collaborative Filtering. Xiaoyuan Su, 2010. Disponível em: <<https://sites.google.com/site/suxiaoyuan/cf>>. Acesso em: 23 set. 2018.
27. WIKIPEDIA, THE FREE ENCYCLOPEDIA. Matrix factorization (recommender systems). Wikipédia, 2018. Disponível em: <[https://en.wikipedia.org/wiki/Matrix_factorization_\(recommender_systems\)](https://en.wikipedia.org/wiki/Matrix_factorization_(recommender_systems))>. Acesso em: 23 set. 2018.
28. HUANG, S.-L. Designing utility-based recommender systems for e-commerce: Evaluation of preference-elicitation methods. Journal Electronic Commerce Research and Applications, p. 398-407, 2011.
29. RICCI, F. Knowledge-Based Recommender Systems. [S.I.].
30. WANG, E. A. A mobile recommendation system based on logistic regression and Gradient Boosting Decision Trees. Conference: 2016 International Joint Conference on Neural Networks, p. 1896-1902, jun. 2016.
31. WIKIPEDIA. Ensemble learning. Wikipedia, 2018. Disponível em: <https://en.wikipedia.org/wiki/Ensemble_learning>. Acesso em: 18 nov. 2018.
32. GASPAR, H. The Cold Start Problem for Recommender Systems. Yuspify, 2015. Disponível em: <<https://www.yuspify.com/blog/cold-start-problem-recommender-systems/>>. Acesso em: 23 set. 2018.
33. INDATA LABS. Approaching the Cold Start Problem in Recommender Systems. Medium, 2015. Disponível em: <<https://medium.com/@InDataLabs/approaching-the-cold-start-problem-in-recommender-systems-e225e0084970>>. Acesso em: 23 set. 2018.

34. WIKIPEDIA. Collaborative Filtering. Wikipedia, 2018. Disponível em: <https://en.wikipedia.org/wiki/Collaborative_filtering>. Acesso em: 23 set. 2018.
35. WIKIPEDIA. Evaluation measures (information retrieval). Wikipedia, 2018. Disponível em: <[https://en.wikipedia.org/wiki/Evaluation_measures_\(information_retrieval\)](https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval))>. Acesso em: 18 nov. 2018.
36. KONOVALOV, V. How can I measure the accuracy of a recommender system? Quora, 2017. Disponível em: <<https://www.quora.com/How-can-I-measure-the-accuracy-of-a-recommender-system>>. Acesso em: 06 nov. 2018.
37. CHAORDIC. CASE DE SUCESSO - HERING. Chaordic. [S.l.], p. 2. 2017.
38. SALESFORCE. Commerce Cloud Einstein. Salesforce. Disponível em: <<https://www.salesforce.com/products/commerce-cloud/commerce-cloud-einstein/>>. Acesso em: 22 set. 2018.
39. SALESFORCE for Retail demo. Intérpretes: Salesforce. [S.l.]: Salesforce. 2018.
40. BAILEY, K. Matrix Factorization with Tensorflow. Kat Bailey, 2016. Disponível em: <<http://katbailey.github.io/post/matrix-factorization-with-tensorflow/>>. Acesso em: 2018 nov. 07.
41. WIKIPEDIA. Netflix Prize. Wikepeia, 2018. Disponível em: <https://en.wikipedia.org/wiki/Netflix_Prize>. Acesso em: 20 nov. 2018.
42. MERCADO & CONSUMO. Grupo Soma investe em desmembramento da Animale e expansão da Farm. Mercado & Consumo, 2018. Disponível em: <<https://www.mercadoeconsumo.com.br/2018/07/18/grupo-soma-investe-em-desmembramento-da-animale-e-expansao-da-farm/>>. Acesso em: 20 nov. 2018.
43. BURKE, R. Knowledge-based recommender systems. Encyclopedia of Library and Information Science, 2000.
44. Z., Z. Fast ML. Evaluating Recommender System, 2015. Disponível em: <<http://fastml.com/evaluating-recommender-systems/>>. Acesso em: 23 set. 2018.
45. GUNAWARDANA, A.; SHANI, G. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. Journal of Machine Learning Research, dez. 2009. 2935-2962.

APÊNDICE A – NOTAS DOS ALGORITMOS DE RECOMENDAÇÃO

	Rankamento	Baseado em conteúdo	Baseado em demografia	Filtragem Colaborativa	Baseado em utilidade	Baseado em conhecimento	Baseado em aprendizado de máquina
1 - Assertividade	1,00	1,00	-	1,00	-	-	1,00
2 - Regras de negócio	0,50	0,50	-	-	0,50	-	0,50
3 - Dados gerais de produtos	0,33	-	0,33	-	-	-	0,33
4 - Dados demográficos	0,25	0,25	0,25	0,25	-	-	0,25
5 - Escalabilidade	0,20	0,20	0,20	-	-	0,20	-
6 - Implementação	0,17	-	-	0,17	-	0,17	-
7 - Estabilidade	0,14	0,14	0,14	-	-	-	0,14
8 - Feedbacks negativos	0,13	0,13	0,13	-	-	-	-
9 - Recomendações explicáveis	0,11	0,11	0,11	-	0,11	0,11	0,11
10 - Integração	0,10	0,10	0,10	0,10	0,10	0,10	0,10
11 - Cold start	0,09	0,09	0,09	0,09	-	0,09	-
Total	1,02	2,34	0,67	1,43	0,49	0,11	1,93

APÊNDICE B – ALGORITMO DE RANKEAMENTO

```
1. def recommend_itens(df, recommendation_method=1 , num_recommendations=5):
2.
3.     if(recommendation_method==1): # recomendação por quantidade de peças vendidas
4.         df_rec = df.groupby(["tipo_produto"], as_index=False)['qtde'].sum()\
5.             .sort_values(by=['qtde'], ascending=False).reset_index(drop=True)
6.         df_rec['Prob_pred'] = (df_rec['qtde']-df_rec['qtde'].min())/
7.             (df_rec['qtde'].max()-df_rec['qtde'].min())
8.         df_rec['Produto'] = df_rec['tipo_produto']
9.
10.    elif(recommendation_method==2): # recomendação por valor de venda
11.        df_rec = df.groupby(["tipo_produto"], as_index=False)['valor'].sum()
12.            .sort_values(by=['valor'], ascending=False).reset_index(drop=True)
13.        df_rec['Prob_pred'] = (df_rec['valor']-df_rec['valor'].min())/
14.            (df_rec['valor'].max()-df_rec['valor'].min())
15.        df_rec['Produto'] = df_rec['tipo_produto']
16.
17.    elif(recommendation_method==3): # recomendação por quantidade de vendas
18.        df_rec = df.groupby(["tipo_produto"], as_index=False)['qtde'].count()\
19.            .sort_values(by=['qtde'], ascending=False).reset_index(drop=True)
20.        df_rec['Prob_pred'] = (df_rec['qtde']-df_rec['qtde'].min())/
21.            (df_rec['qtde'].max()-df_rec['qtde'].min())
22.        df_rec['Produto'] = df_rec['tipo_produto']
23.
24.    #retorno na forma de um pandas dataframe
25.    return df_rec.loc[:num_recommendations, :][['Produto', 'Prob_pred']]
```

APÊNDICE C – ALGORITMO BASEADO EM SIMILARIDADES DE PRODUTO

```
1. #Função que obtém a matrix de similaridade dos produtos
2. def get_matrix_rec(df):
3.     df_products = pd.DataFrame(df.tipo_produto.unique())
4.     df_products.rename(inplace=True, columns={0: "tipo_produto"})
5.     count = CountVectorizer()
6.     count_matrix = count.fit_transform(df_products['tipo_produto'])
7.     cosine_sim = cosine_similarity(count_matrix, count_matrix)
8.     df_rec = df.groupby(['codigo_cliente', "tipo_produto"], as_index=False)['qtde'].sum(
9.         )
10.    df_rec = df_rec.sort_values(by=['qtde'], ascending=False).reset_index(drop=True)
11.    #retorna matriz de similaridade, matriz com os produtos e matriz com as compras
12.    return df_products, cosine_sim, df_rec
13.
14. #Função que busca produtos similares ao buscado dentro da matriz de similaridade
15. def get_recommendations_similar_products(df_products, product, cosine_matrix,
16.     factor =1, products=10):
17.     indices = pd.Series(df_products.index, index=df_products.tipo_produto)
18.     idx = indices[product]
19.     #busca as similaridades dentro da matriz de similaridade
20.     sim_scores = list(enumerate(cosine_matrix[idx]))
21.     sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
22.     sim_scores = sim_scores[0:(products+2)]
23.     product_indices = [i[0] for i in sim_scores]
24.     product_similarity = [i[1] for i in sim_scores]
25.     df_to_return = pd.DataFrame(df_products['tipo_produto'].iloc[product_indices]).r
26.     eset_index(drop=True)
27.     df_to_return['similarity'] = product_similarity
28.     df_to_return['factor'] = factor
29.     #retorna a similaridade do produto e um fator que será usado para ponderar as re
30.     comendações
31.     return df_to_return
32.
33. #Função que retorna as recomendações para um cliente, seguindo os produtos que ele j
34.     á comprou
35. def recommend_itens(df_rec, df_products, codigo_cliente, cosine_matrix, num_recom
36.     mendations=5 ):
37.
38.     size_df_rec = 5
39.     df_rec = df_rec.loc[df_rec.codigo_cliente==(codigo_cliente)]
40.     df_rec = df_rec.sort_values(by=['qtde'], ascending=False).reset_index(drop=True)
41.
42.     df_rec = df_rec.loc[:(size_df_rec-1), :]
43.     df_rec_prod = pd.DataFrame()
44.     #busca produtos similares àqueles que ele já comprou
45.     for row in df_rec.iterrows():
46.         df_rec_prod = df_rec_prod.append(get_recommendations_similar_products(
47.             df_products, row[1][1],
48.             cosine_matrix, (size_df_rec - row[0])), ignore_index=True)
49.     #pondera as similaridades dos produtos seguindo os mais comprados pelo cliente
50.     df_rec_prod['peso'] = df_rec_prod['similarity']*df_rec_prod['factor']
51.     df_rec_prod = df_rec_prod.groupby(["tipo_produto"], as_index=False)['peso'].
52.     sum()
53.     df_rec_prod = df_rec_prod.sort_values(by=['peso'], ascending=False).
54.     reset_index(drop=True)
55.     df_rec_prod['Prob_pred'] = (df_rec_prod['peso']-df_rec_prod['peso'].min())/
56.     (df_rec_prod['peso'].max()-df_rec_prod['peso'].min())
57.     df_rec_prod['Produto'] = df_rec_prod['tipo_produto']
58.     df_rec_prod['codigo_cliente'] = codigo_cliente
59.     #Retorno das recomendações na forma de um pandas dataframe
60.     return df_rec_prod[0:(num_recommendations)][['Produto', 'Prob_pred', 'codigo_clie
61.     nte']]
```

APÊNDICE D – ALGORITMO DE FILTRAGEM COLABORATIVA

```
1. #Função que faz a fatoração da matriz de usuários x produtos
2. def matrix_factorization(ratings, k=40):
3.     R = ratings.values
4.     user_ratings_mean = np.mean(R, axis = 1)
5.     R_demeaned = R - user_ratings_mean.reshape(-1, 1)
6.     #realiza a fatoração em k fatores
7.     U, sigma, Vt = svds(R_demeaned, k )
8.     sigma = np.diag(sigma)
9.     #retorna os parametros S, V e D da fatoração
10.    return U, sigma, Vt
11.
12. #faz a estimativa de avaliação para todos os usuários
13. def predict_to_all_users(U, sigma, Vt, ratings):
14.     R = ratings.values
15.     user_ratings_mean = np.mean(ratings.values, axis = 1)
16.     all_user_predicted_ratings = np.dot(np.dot(U, sigma), Vt) + user_ratings_mean.re
shape(-1, 1)
17.     preds_df = pd.DataFrame(all_user_predicted_ratings, columns = ratings.columns)
18.     df = pd.merge(pd.DataFrame(ratings.reset_index()['codigo_cliente']), preds_df,
left_index=True, right_index=True , how='inner')
19.     #retorna a matriz de usuários x produtos reconstruída
20.     return df
21.
22. #busca dentro da matriz reconstruída de usuários x produtos os produtos
23. # que teriam as maiores avaliações pelo usuário
24. def recommend_itens(df_pred, codigo_cliente, ratings_user, num_recommendations=5):
25.     user_predictions = df_pred.loc[df_pred.codigo_cliente==(codigo_cliente)]
26.     user_predictions = user_predictions.T[1:]
27.     user_predictions = user_predictions.reset_index()
28.     user_predictions.rename(columns = {'index':'Produto',user_predictions.columns[1]
:'Pred_rating' },inplace=True)
29.     ratings_user = ratings_user.loc[ratings_user.codigo_cliente==(codigo_cliente)].T
[1:].reset_index()
30.     ratings_user = ratings_user.rename(columns = {'index':'Produto',ratings_user.col
umns[1]:'Valor' })
31.     ratings_user = ratings_user[ratings_user.Valor!=0]
32.     user_predictions2 = pd.merge(user_predictions, ratings_user, on='Produto', how=
'left')
33.     user_predictions2.fillna(0,inplace=True)
34.     user_predictions2['Valor'] = user_predictions2['Valor'].astype(int)
35.     user_predictions2['Já comprado?'] = user_predictions2['Valor'].apply(lambda x: T
rue if x!=0 else False)
36.     user_predictions2['Prob_pred'] = (user_predictions2['Pred_rating']-
user_predictions2['Pred_rating'].min())/ \
37.         (user_predictions2['Pred_rating'].max()-
user_predictions2['Pred_rating'].min())
38.     user_predictions2.drop(['Valor','Pred_rating'], axis=1, inplace=True)
39.     user_predictions2 = user_predictions2.sort_values(by=['Prob_pred'], ascending=Fa
lse )\
40.         [:num_recommendations].reset_index(drop=True)
41.     user_predictions2['codigo_cliente'] = codigo_cliente
42.     #retorna recomendações para um cliente em específico
43.     return user_predictions2
```

APÊNDICE E – ALGORITMO DE ESCOLHA FINAL

```
1. clf_compra = GradientBoostingClassifier(n_estimators=10, learning_rate=0.5, max_dept
   h=3, random_state=0)
2. clf_compra.fit(train_compra[features], train_compra['compra'])
3. preds_compra = clf_compra.predict(test[features])
4. preds_prob_compra = clf_compra.predict_proba(test[features])
5.
6. clf_neutro = GradientBoostingClassifier(n_estimators=10, learning_rate=0.5, max_dept
   h=3, random_state=0)
7. clf_neutro.fit(train_neutro[features], train_neutro['neutro'])
8. preds_neutro = clf_neutro.predict(test[features])
9. preds_prob_neutro = clf_neutro.predict_proba(test[features])
10.
11. clf_devolucao = GradientBoostingClassifier(n_estimators=15, learning_rate=0.5, max_d
   epth=2, random_state=0)
12. clf_devolucao.fit(train_devolucao[features], train_devolucao['devolucao'])
13. preds_devolucao = clf_devolucao.predict(test[features])
14. preds_prob_devolucao = clf_devolucao.predict_proba(test[features])
```

APÊNDICE F – RECOMENDAÇÃO DE PRODUTOS COM REGRAS DE NEGÓCIO

```
1. def top_produtos_regra_negocio(produtos, regra, tipo_produto, filtro, num_rec = 2):
2.     produtos['grupo'] = produtos['grupo'].str.upper()
3.     produtos['preco_varejo_original'] = produtos['preco_varejo_original'].map(float)
4.     produtos['subgrupo'] = produtos['subgrupo'].str.upper()
5.     produtos['colecacao'] = produtos['colecacao'].str.upper()
6.     produtos['linha'] = produtos['linha'].str.upper()
7.     produtos['estilista'] = produtos['estilista'].str.upper()
8.     filtro = "(id_produto>1 & (qtde>1) ) & (" + filtro + ")"
9.     produtos = produtos[produtos.tipo_produto==tipo_produto]
10.    produtos = produtos.query(filtro)
11.    produtos = produtos.dropna()
12.    if(produtos.shape[0]>0):
13.
14.        if(regra==1): #maior quantidade nos ultimos meses
15.            df_final = produtos.sort_values(by=['qtde'], ascending=False)[:num_rec]
16.
17.        elif(regra==2): #maior margem de lucro
18.            produtos['mergem_lucro'] = produtos['preco_custo']/produtos['preco_varej
19.            o_original']
20.            df_final = produtos.sort_values(by=['mergem_lucro'], ascending=False)[:n
21.            um_rec]
22.
23.        elif(regra==3): #maior estoque disponível (não implementada)
24.            raise ValueError('Regra não implementada devido a disponibilidade de dad
25.            os')
26.        else:
27.            return pd.DataFrame()
28.            #raise ValueError('Sem correspondência de produtos')
29.            return df_final[['desc_produto', 'id_produto', 'id_marca_linx', 'grupo', 'subgrupo',
30.            'colecacao', 'linha', 'estilista', 'preco_varejo_original', 'tipo_produto']]\
31.            .reset_index(drop=True)
```